LA REPUBLIQUE TUNISIENNE
MINISTERE DES TECHNOLOGIES DE LA COMMUNICATION

Guide technique d'accessibilité pour la création et la refonte des sites Web de l'administration publique

Date : Juillet 2010
Version 1.0

I- **Mode d'emploi du présent guide technique** 5

II- Recommandations pour le développement de l'interface de nouveaux sites 7

1. Structure - Présentation - Action 8

2. Code (x)HTML 11

a. Frame 13

b. Internationalisation 15

c. Attribut LANG 16

d. Attribut DIR 17

e. Attribut CHARSET 19

f. Tag (x)HTML pour les contenus 20

g. Actualisation et redirection 24

3. CSS (Cascading Style Sheet) 25

4. Script et autres objets de programmation 28

a. Alternatives aux scripts, applets ou autres objets de programmation 29

b. Indépendance du dispositif d'input 30

c. Fenêtres et pop-up 32

5. Lay-out des pages 34

a. Caractéristiques générales 34

b. Taille relative du lay-out 38

c. Compatibilité avec les différents navigateurs 40

6. Présentation des contenus 41

a. Organisation des titres 41

b. Style rédactionnel 43

c. Style graphique 44

d. Numéros et caractères spéciaux 46

7. Navigation et liens 46

a. Caractéristiques générales 46

b. Reconnaissance du contexte 46

c. Activation et caractère significatif des liens 46

d. Caractère reconnaissable des liens 46

e. Skip link 46

f. Espaces entre les liens 46

8. Formulaires 46

a. Caractéristiques générales 46

b. Étiquettes (label) 46

c. FIELDSET et LEGEND, OPTGROUP 46

d. Boutons 46

e. Messages 46

f. Durée des sessions 46

g. Captcha 46

9. Tableaux de données 46

a. Caractéristiques générales 46

b. Titre des tableaux et des colonnes 46

c. En-têtes multiples 46

10. Graphisme 46

a. Caractéristiques générales 46

b. ALT 46

c. LONGDESC 46

d. Couleurs et contrastes 46

e. Plans 46

11. Documents non html 46

a. PDF 46

b. Flash 46

c. Fichiers audio/vidéo 46

III- vérification d'accessibilité basée sur la check-list 46

1. Check-list d'accessibilité 46

2. Outils d'aide à l'évaluation 46

a. Navigateur 46

b. Outils d'assistance technologique 46

c. Outils d'évaluation automatique du code 46

3. Application de la check-list 46

I- Mode d'emploi du présent guide technique

Le présent document contient les lignes directrices non seulement pour le développement d'interfaces accessibles des nouveaux sites et services web, mais aussi pour la mise en conformité des sites et services déjà existants fournis par l'administration publique.

Dans ce contexte, le terme d'accessibilité désigne la capacité des systèmes informatiques à fournir des services et des informations accessibles, sans discriminations, visant tous les utilisateurs, indépendamment de la présence ou non d'handicaps (physiques, sensoriels, cognitifs) et des spécificités matérielles et logicielles. Le concept d'accessibilité limité à la possibilité d'accès aux informations est englobé par le concept d'accessibilité vue comme « possibilité de bénéficier ». Dans cette optique, il est pleinement assimilable à celui d'utilisabilité, entendue comme la capacité du logiciel à permettre à des utilisateurs spécifiques d'exécuter des activités particulières avec efficacité, efficience et satisfaction dans un contexte d'utilisation déterminé.

L'objectif est de fournir à l'utilisateur toutes les fonctionnalités dont il a besoin pour exécuter les activités de façon correcte et rapide, en relation avec les objectifs d'interaction à atteindre, le contexte technologique et organisationnel dans lequel il opère, les exigences qu'il manifeste, quel que soit l'outil d'interaction spécifique utilisé, qu'il s'agisse d'un navigateur, d'un outil d'assistance technologique ou d'un PDA.

Le présent document est composé de deux parties :

- le chapitre 2 contient les recommandations techniques pour la réalisation d'interfaces utilisables et accessibles.
- le chapitre 3 contient une méthodologie visant à évaluer le respect des recommandations indiquées dans le chapitre 2.

Les recommandations d'accessibilité et d'utilisabilité contenues au chapitre 2 se basent sur la législation et les directives internationales et nationales (WCAG 1.0, WCAG 2.0, ISO 9241, Référentiel Général d'Accessibilité pour les Administrations tunisiennes), les meilleures pratiques du secteur, la littérature spécialisée et les nouvelles tendances dans le secteur web.

L'élaboration de ces recommandations a tenu compte de certains critères importants.

D'une part, elles représentent la synthèse des meilleures pratiques et recommandations indiquées dans la littérature et dérivant de l'expérience pratique : elles englobent et dépassent les spécificités des directives considérées individuellement.

D'autre part, tout en garantissant l'accessibilité technique, ces recommandations visent de façon explicite à rendre réellement accessibles les informations et services mis à la disposition des citoyens, indépendamment de leurs spécificités physiques, des préférences et des instruments utilisés pour interagir avec le web.

Enfin, en se focalisant sur la séparation entre contenus, présentation et action, elles facilitent la satisfaction d'autres facteurs de qualité du logiciel, comme la portabilité, la maintenabilité ou l'efficacité.

Il convient également de noter que le présent document, adopte une approche pragmatique, basée sur les objets de programmation, et non une approche théorique basée sur des principes de conception.

Ceci permet au développeur de trouver toutes les lignes directrices relatives à un sujet ou à un objet à l'intérieur d'un même chapitre, sans devoir les rechercher parmi les différents principes d'accessibilité.

Il est toutefois important de rendre également explicite le principe auquel ces recommandations font référence; dès lors, pour chaque objet, le document présentera les lignes directrices du W3C/WAI dont les recommandations indiquées s'inspirent.

De plus, lorsque cela est jugé nécessaire, le document fournit :

- des exemples d'application pratique des recommandations (comment écrire le code);
- le type de contrôle à appliquer pour vérifier l'application correcte des recommandations;
- la manière d'effectuer le contrôle;
- les difficultés du contrôle;
- l'indication des objets de la check-list d'accessibilité (mentionnée au chapitre 3) dans laquelle il y a le résultat du contrôle.

Par ailleurs, le chapitre 3 aborde l'aspect de la vérification. Il illustre une méthodologie qui permet à l'expert en accessibilité d'exprimer un avis sur la conformité ou la non-conformité des pages réalisées aux recommandations d'utilisabilité et d'accessibilité contenues dans le chapitre 2.

Cette méthodologie se base sur l'utilisation de différents instruments d'évaluation, comme le navigateur, les technologies d'assistance, les outils pour l'évaluation automatique et semi-automatique du code.

Le résultat du contrôle effectué avec ces instruments est repris dans la check-list jointe à l'annexe 1, en indiquant pour chaque item s'il est conforme (Oui), s'il n'est pas conforme (Non) ou s'il n'est pas applicable (N/C) après avoir effectué le contrôle afférent.

L'évaluation sera positive seulement dans le cas où tous les items auront une réponse positive ou N/C.

II- Recommandations pour le développement de l'interface de nouveaux sites

Le présent chapitre contient les recommandations pour l'organisation et la réalisation du graphisme, de la structure des pages, de la navigation et de la structure des contenus, c'est-à-dire de tout ce qui doit être organisé et réalisé au niveau du code pour qu'un site/service web soit :

- accessible et utilisable selon les définitions du Cahier des Charges Type ;
- indépendant de l'instrument spécifique d'interaction employé par l'utilisateur et donc compatible avec toutes les technologies et notamment celles d'assistance utilisées par les personnes handicapées, comme indiqué au Cahier des Charges Type ;
- conforme aux principes d'utilisabilité et d'accessibilité indiqués par la littérature spécialisée.

Ces recommandations sont rédigées en suivant les différentes directives nationales et selon les meilleures pratiques existantes au niveau national et international. Elles ont été réunies en considérant certains aspects fondamentaux :

- elles dépassent les spécificités individuelles de chaque directive en les englobant toutes ;
- elles prévoient et garantissent l'accessibilité technique et en même temps, ces recommandations visent de façon explicite une exploitabilité réelle des informations et des services mis à la disposition des citoyens, indépendamment de leurs particularités physiques, des préférences et des instruments utilisés pour interagir avec le web, qu'il s'agisse d'instruments communs ou d'outils d'assistance technologique ;
- les sites et les services web développés selon les présentes recommandations continueront à fonctionner quand les navigateurs traditionnels évolueront et que des dispositifs d'accès à internet nouveaux et plus modernes feront leur apparition sur le marché.

Lors de la rédaction des recommandations, il a été tenu compte des législations, directives, normes et lignes directrices suivantes :

- ISO 9241 « Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV) »
- Web Accessibility Initiative (WAI) (http://www.w3.org/WAI/)
- Web Content Accessibility Guidelines 1.0 et 2.0 : http://www.w3.org/TR/WCAG10/

http://www.w3.org/TR/WCAG20/

- W3C Internationalization (I18n) Activity (http://www.w3.org/International/)
- Référentiel Général d'Accessibilité pour les Administrations : http://www.references.modernisation.gouv.fr/rgaa-accessibilite
- Guide Accessi Web : http://www.accessiweb.org/fr/guide_accessiweb/index.html
- Guide BrailleNet à l'usage des webmestres : http://www.braillenet.org/accessibilite/guide/
- Section 508 des États-Unis (http://www.section508.gov/)
- Décret du Ministère de l'innovation et des technologies du 8 juillet 2005 sur « les conditions techniques et les différents niveaux d'accessibilité aux outils informatiques » : http://www.pubbliaccesso.gov.it/normative/DM080705.htm
- Jacob Nielsen's Usability Heuristics : http://www.useit.com/papers/heuristic/heuristic_list.html
- Web Style Guide by Patrick J. Lynch and Sarah Horton : http://www.webstyleguide.com/
- Research-based Web Design and Usability Guidelines by U.S. Department of Health and Human Services : http://www.usability.gov/pdfs/guidelines.html
- IBM Accessibility Developer Guidelines : http://www-03.ibm.com/able/guidelines/index.html

Les présentes recommandations sont destinées à un personnel technique disposant de connaissances minimales de HTML, XHTML, CSS et JavaScript, ainsi que les web designers et les développeurs web.

1. Structure - Présentation - Action

Pour garantir une meilleure accessibilité, c'est-à-dire une meilleure interprétation des contenus par les outils d'assistance technologique, il est important d'adopter des techniques particulières de réalisation des pages web, comme le W3C le recommande dans la Web Accessibility Initiative (WAI).

Selon les indications du W3C/WAI, chaque page est composée de trois éléments distincts, comme

le montre l'illustration suivante :

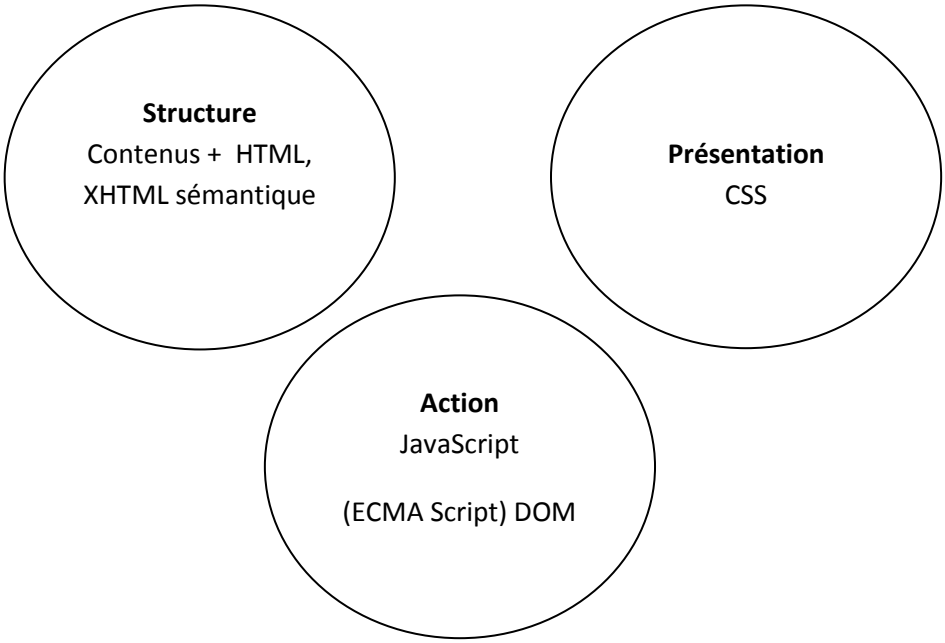


Figure 1 : Eléments d'une page web

La **structure** est formée des parties indispensables à une page web :

- contenus ;
- marquage sémantique (HTML, XHTML).

Il est fondamental que la page soit bien structurée et codifiée en respectant la signification sémantique des marqueurs. En d'autres termes, il est important que les données textuelles soient formatées selon leur signification structurelle propre :

- en-tête ;
- en-tête secondaire ;
- paragraphe ;
- liste numérotée ;

- liste de définitions ;
- et autres.

L'utilisation du code HTML ou XHTML sans erreurs, sans tags indésirables, produira une page web complètement portable, compréhensible par tous les navigateurs web, par les outils technologiques d'assistance, par les navigateurs textuels, par les téléphones portables de nouvelle génération, etc.

La **présentation** est définie par les feuilles de style CSS (Cascading Style Sheets). Elle représente et codifie le style graphique qui est donné au contenu depuis le formatage de la page web jusqu'à la disposition des objets et des contenus, en passant par la typologie et la taille des polices de caractères et par les couleurs utilisées.

Dans la plupart des cas, la présentation concerne la manière dont un document est visualisé avec un navigateur, mais elle peut aussi affecter l'impression, la synthèse vocale, et même la présentation graphique sur un PDA ou un portable.

Du point de vue technique, il est conseillé de séparer dans des fichiers différents la structure et la présentation.

Cette modalité de travail, où les CSS sont codifiés sur un fichier à part, séparé du fichier de la structure, permet de changer un composant sans influer sur l'autre. Il est possible, par exemple, d'appliquer la même modalité de présentation à de nombreuses pages ou d'effectuer des modifications graphiques à des textes et des liens (changer de couleur ou de dimension du caractère, etc.) sans toucher le lay-out de structure.

Le concept **d'action**, représenté par JavaScript ou d'autres objets de programmation, se réfère au « comportement » du site, aux effets dynamiques et aux animations que l'on entend insérer dans les différentes pages.

L'action est réalisée à travers l'adoption de ECMAScript (version standard de JavaScript) et l'utilisation de W3C DOM. Comme pour la structure et la présentation, il convient de codifier l'action dans un fichier séparé avec l'extension .js.

Il est conseillé d'utiliser des techniques qui permettent de projeter des pages où les fonctionnalités fondamentales sont utilisables également sans Javascript. De cette manière, le site fonctionne, même dans le cas où la technologie appliquée par l'utilisateur ne supporte pas Javascript. L'amélioration progressive (progressive enhancement) fait précisément référence à cela : une amélioration progressive de la page, à partir des fonctionnalités de base auxquelles tous peuvent accéder.

En plus d'une meilleure accessibilité, la séparation « structure – présentation – action » décrite ci-dessus offre aussi d'autres avantages techniques importants :

- Compatibilité descendante : les pages sont rendues résistantes dans le temps, ce qui limite le risque que les navigateurs et technologies futurs ne soient plus en mesure d'interpréter le code utilisé;
- Téléchargement et visualisation des pages plus rapides : en réalisant des fichiers de dimensions plus contenues et avec moins de code, les délais de visualisation se réduisent ;
- Meilleur positionnement dans les moteurs de recherche : la séparation entre structure/contenus et présentation fait en sorte que le contenu est la partie la plus consistante dans la taille finale des fichiers ; cet aspect, associé au marquage sémantique du code, améliore le positionnement dans les moteurs de recherche ;
- Portabilité : un document marqué de manière opportune peut être facilement adapté aux dispositifs de navigation alternatifs comme les PDA, téléphones portables, assistants particuliers pour les personnes handicapées.

Il y a aussi des avantages en termes de coûts et de qualité du produit, tels que :

- Meilleure qualité du code : le respect du marquage standard augmente la qualité des produits réalisés ;
- Moindre coût d'entretien et de développement : utiliser un code bien structuré et sémantiquement correct réduit considérablement les délais et les coûts de création et d'entretien du site ;
- Economie de la bande : les fichiers peuvent peser jusqu'à 50% moins, ce qui réduit les délais de réponse du serveur, avec par conséquent, une demande de bande diminuée.

2. Code (x)HTML

L'HTML est un langage de marquage qui décrit les mécanismes de représentation (structurels, sémantiques ou de présentation) du texte, en utilisant des conventions standardisées, du domaine public, dont la syntaxe est établie par le W3C.

Au cours du temps, le W3C a reformulé l'HTML selon les standards de l'XML, définissant ainsi un nouveau langage de marquage appelé XHTML. Les règles d'écriture d'une page XHTML sont contenues dans un ensemble de directives (recommandations) publiées depuis le 26 janvier 2000 par la W3C (http://www.w3.org/TR/xhtml1/).

Dans le présent document, nous entendons par « (x)HTML » indifféremment HTML ou XHTML.

DOCTYPE

La première étape pour arriver à un code (x)HTML de qualité consiste à écrire le code des pages de manière conforme aux standards web officiels, comme indiqué par les recommandations du W3C.

Le DOCTYPE doit être le premier élément à ouvrir le document (x)HTML et il doit comporter l'indication de la DTD (Document Type Declaration) à laquelle la page est conforme.

Il existe 3 types de DTD, et autant pour le code HTML et XTML :

- Strict : c'est la DTD la plus rigoureuse dans l'écriture du code parce qu'elle prévoit une nette séparation entre la structure et la présentation, avec par conséquent l'utilisation de l'HTML ou XHTML sémantiques pour la structure et des CSS pour la présentation graphique ;
- Transitional : c'est la DTD la plus utilisée, elle comprend des tags et des attributs indésirables, c'est-à-dire utilisés dans des anciennes versions de l'HTML, et à caractère de présentation ;
- Frameset : elle est identique à la DTD Transitional, mais elle est utilisée quand une page est subdivisée en cadres.

Les outils d'assistance technologique travaillent mieux si les pages web sont réalisées avec la séparation structure – présentation – action, les DTD les plus indiquées pour obtenir des pages accessibles sont les plus rigoureuses, puisqu'elles imposent cette séparation, à savoir :

- DTD Strict de HTML 4.01 ;
- DTD Strict de XHTML 1.0.

Pour les DTD de type Transitional, il est de toute façon nécessaire :

- D'utiliser les CSS pour régler les caractéristiques de présentation et de style de la page ;
- D'éviter l'insertion de tags indésirables, parce que les outils d'assistance technologique ne les interprètent pas correctement ;
- D'éviter l'ouverture de nouvelles fenêtres du navigateur, avec l'attribut <TARGET="_BLANK">, parce que certaines personnes handicapées et valides avec une faible expérience dans l'utilisation des technologies informatiques peuvent ne pas avoir la perception de cet événement et donc rester bloquées ;
- Ne pas utiliser les images pour gérer les espaces et les marges, parce que celles-ci alourdissent inutilement la page, mais elles créent aussi des problèmes avec les outils d'assistance technologique.

Exemple de la manière d'indiquer correctement la DTD :

- pour DTD Strict de HTML 4.01 :
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
- pour DTD Strict de XHTML 1.0 :
  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
- pour DTD Transitional de HTML 4.01 :
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
- pour DTD Transitional de XHTML 1.0 :
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">.

Référence WCAG 2.0 : 4.1.1

Type de contrôle :

☒ Test automatique

☐ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 1 ; 2

a. Frame

Les frames sont de vieilles structures (x)HTML que l'on tend à ne plus utiliser parce qu'ils présentent certains inconvénients. Il n'est pas possible d'ajouter un marque-page directement à la page d'intérêt spécifique, ils sont indiqués comme des pages séparées dans les moteurs de recherche, l'impression du contenu peut être difficile, etc.

Il est conseillé de ne plus utiliser les frames pour la réalisation de nouveaux sites/applications.

Pour les sites/applications déjà réalisés avec des frames, il est conseillé de migrer vers la version avec DTD Transitional, voir DTD Strict.

Dans l'intervalle, pour garantir l'accessibilité, il est nécessaire de modifier les pages qui contiennent les frames en insérant la DTD Frameset et en respectant les indications suivantes :

- Eviter de spécifier les caractéristiques de présentation et de style à l'intérieur de

la page et recourir par contre aux CSS pour obtenir le même effet graphique ;

- Définir les lignes et les colonnes de subdivision du frameset, avec des mesures relatives (par exemple, 20%, 80%), de manière à permettre le redimensionnement proportionnel de la page ;
- Insérer à l'intérieur de <NOFRAMES> ... </NOFRAMES> les liens vers les différentes pages qui composent le frameset de manière à permettre la navigation individuelle. Les liens doivent être significatifs, de manière à faire comprendre clairement à l'utilisateur à quoi ils renvoient, et ce que contient le fichier qu'ils ouvrent (exemple : « Menu du site » et non « Frame latéral ») ;
- Dans la page frameset :
- Utiliser l'attribut TITLE dans le tag <FRAMESET> et dans le tag <FRAME>;
- Utiliser dans la section <HEAD> ... </HEAD> le tag <TITLE>.
- Dans les pages insérées dans le frameset, spécifier le tag <TITLE> dans <HEAD> ... </HEAD>.
- Afin d'assurer une interaction efficace, la description insérée dans le <TITLE> doit être significative.

Exemple de frame accessible :

Pour rendre accessibles les pages web avec frame, il est nécessaire de spécifier des <TITLE> significatifs et la taille relative des colonnes du frameset.

- Frameset:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<HEAD>
<TITLE> Accessibilité des frame</TITLE>

</HEAD>

<FRAMESET TITLE="Menu de la page des nouveautés du site du Département des politiques fiscales"
cols="23%,*">
<FRAME TITLE="Aire menu du site" SRC="frame02a.htm">
<FRAME TITLE="Aire nouveautés du site" SRC="frame02b.htm">
<NOFRAMES>
<a HREF="frame02a.htm">Menu du site du Département des politiques fiscales</a><BR><BR>
<a HREF="frame02b.htm">Page des nouveautés du site du Département des politiques fiscales</a>
</NOFRAMES>
</FRAMESET>
```

- File frame02a.htm:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE> Menu du site du Département des politiques fiscales </TITLE>
</HEAD>
<BODY>
<p><a HREF="#">Entrée de menu 1</a></p>
<p><a HREF="#"> Entrée de menu 2 </a></p>
</BODY>
</HTML>
```

- File frame02b.htm:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE> Page des nouveautés du site du Département des politiques fiscales</TITLE>
</HEAD>
<BODY>
<H1> Nouveautés du site du Département des politiques fiscales</H1>
<p>Philosophiae artibus ipse ....</p>
</BODY>
</HTML>
```

Exemple de iframe accessible :

```
<iframe src="news.html" id="testiframe"
    name="testiframe" title="news">
    <a href="news.html">News</a>
</iframe>
```

Référence WCAG 2.0 : 2.4.1

Type de contrôle :

☒ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 3

b. Internationalisation

Par le mot « internationalisation », le W3C Internationalization (I18n) Activity désigne les technologies qui permettent un développement correct d'un site/application en rapport avec les caractéristiques culturelles et linguistiques d'une cible d'utilisateurs.

Internationalisation est souvent écrit « i18n », où 18 est le nombre de lettres entre le « i » et le

« n » dans le mot anglais (internationalization).

Les trois attributs de l'(x)HTML qui permettent l'internationalisation sont :

- LANG (language), qui identifie la langue du texte ;
- DIR (direction), pour la directionnalité du texte ;
- CHARSET (character set), pour le codage des caractères.

Ces trois attributs sont particulièrement importants quand il faut développer un site multilingue. En particulier l'attribut DIR, non utilisé dans les sites qui utilisent des langues écrites de gauche à droite, est par contre obligatoire pour les sites en langue arabe ou dans d'autres langues écrites de droite à gauche.

c. Attribut LANG

Immédiatement après avoir défini la DTD, avec l'élément <DOCTYPE>, il faut toujours spécifier la langue principale de la page à travers l'attribut LANG de l'élément <HTML> :

Ceci permet aux lecteurs d'écran les plus récents de régler automatiquement le synthétiseur vocal sur la langue dans laquelle est écrit le document, évitant à l'utilisateur de devoir effectuer la manœuvre demandée.

Exemple de la manière de spécifier l'attribut LANG au niveau de la page :

- pour HTML**:** <html lang="fr">
- pour XHTML**:** <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr")

Référence WCAG 2.0 : 3.1.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 4

Si, à l'intérieur de la page, il y a des textes d'une certaine longueur (phrases de plusieurs lignes)

écrits dans une langue différente de la langue principale, il peut être utile de spécifier la langue avec laquelle le lecteur d'écran doit lire ce texte spécifique. L'attribut à utiliser LANG doit être inséré dans le tag principal.

Il ne faut pas utiliser le changement de langue pour des mots individuels, des mots étrangers entrés dans la langue courante ou des acronymes parce que :

- le lecteur d'écran interrompt la lecture pour charger le nouveau vocabulaire, ce qui rend le texte peu compréhensible ;
- certains mots peuvent ne pas être reconnus s'ils sont lus dans la langue originale (par exemple, HTML serait lu « eytch tii em el » et pourrait ne pas être compris pour qui ne parle pas couramment l'anglais).

Exemple de la manière de spécifier l'attribut LANG dans le texte :

 La Divine Comédie, de Dante, commence de cette façon : Nel mezzo del cammin di nostra vita mi ritrovai per una selva oscura, che la diritta via era smarrita. Le grand poète qu'il voulait dire …

Référence WCAG 2.0 : 3.1.2

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 22

d. Attribut DIR

En plus de spécifier la langue d'un document, il peut être nécessaire de spécifier la directionnalité de base (gauche-droite ou droite gauche) de tout le texte ou d'une partie seulement de celui-ci.

Ceci se fait à travers l'attribut DIR. La valeur prédéfinie de l'attribut DIR est « ltr » (left-to-right, texte de gauche à droite).

Il faut se rappeler que, en insérant l'attribut DIR="rtl" dans l'élément <HTML>, dans les différentes versions d'Internet Explorer, la barre de déroulement du navigateur se positionne à gauche.

Exemple de la manière de spécifier l'attribut DIR au niveau de la page :

- pour HTML **:** <html lang="fr" dir="ltr">
- pour XHTML **:** <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr" dir="ltr")

Référence WCAG 2.0 : 1.3.2 ; 3.1.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 4

Si, à l'intérieur de la page, il y a des textes spécifiques écrits dans une langue ayant une directionnalité de lecture différente de la principale, il faut préciser la direction du texte pour le visualiser et le lire correctement.

Dans ce cas, la directionnalité du texte peut être insérée dans le tag principal (ex.: ou) ou reproduite dans le CSS au moyen de la propriété direction (ex.: p { direction: rtl }).

Exemple de la manière de spécifier l'attribut DIR dans le texte :

L'expression تكنولوجيا المعلومات للجميع en arab signifie ….

Référence WCAG 2.0 : 3.1.2

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 22

e. Attribut CHARSET

Tous les documents (x)HTML doivent spécifier leur codage des caractères.

À l'intérieur de la section <HEAD>, le meta tag CHARSET sert à définir la table de caractères utilisés dans une page web.

Le terme Charset, abréviation de Character set, indique un ensemble de caractères (Unicode, ISO-8859-1, ASCII) où chaque caractère est associé à un numéro de manière univoque. Cette association permet au navigateur une représentation correcte du texte.

D'habitude, un document (x)HTML est écrit en utilisant la table de caractères ISO qui, par rapport au codage ASCII, permet l'utilisation de caractères spéciaux comme les lettres accentuées.

Dans le cas de sites multilingues, une norme utile consiste à utiliser un codage UTF-8, dans la mesure où :

- Il élargit la gamme des caractères supportés, comme la lettre grecque bêta (β) pour les formules mathématiques, les caractères graphiques (•, ⍰), les symboles commerciaux (™, ©, ®) ;
- Il simplifie de nombreux aspects de l'internationalisation sur le web ;
- Il est largement supporté par tous les navigateurs les plus récents.

Exemple pour ISO :

- pour HTML : <meta http-equiv="content-type" content="text/html; charset= iso-8859-1">
- pour XHTML : <meta http-equiv="content-type" content="text/html; charset= iso-8859-1"/>

Exemple pour UTF-8 :

- pour HTML : <meta http-equiv="content-type" content="text/html; charset= UTF-8">
- pour XHTML : <meta http-equiv="content-type" content="text/html; charset= UTF-8"/>

Référence WCAG 2.0 : 3.1.1 ; 3.1.2

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 4

f. Tag (x)HTML pour les contenus

Titres

Une page web doit être structurée en titres, sous-titres, chapitres, sous-chapitres et ainsi de suite.

Certains outils d'assistance technologique, comme les lecteurs d'écran, utilisent les titres pour donner une vision d'ensemble de la page web en définissant la structure du document. Si les textes des titres sont clairs et compréhensibles, leur lecture sera aisée et rapide, les non-voyants pourront se déplacer rapidement parmi les contenus de la page.

Le titre principal de la page doit être codifié avec <H1>, <H2> pour les titres de deuxième niveau, <H3> pour les titres de troisième niveau, etc., jusqu'à <H6>.

Il ne faut jamais utiliser les tags <H1> ... <H6> dans un but purement décoratif, par exemple uniquement pour agrandir la taille du texte. La caractérisation graphique de chaque type de titre (format, couleur, grandeur, éventuelle indentation, espacement, etc.) doit se faire via les CSS.

Pour un approfondissement sur la manière de structurer les titres à l'intérieur d'une page web, nous renvoyons aux paragraphes II.5, II.6.a et II.7.a.

Exemple de la manière d'organiser les titres <H1> … <H6> :

```
<head>
<title>Cooking</title>
</head>
<body>
<h1>Cooking techniques</h1>
... some text here ...
<h2>Cooking with oil</h2>
... text of the section ...
<h3>Sautéeing</h3>
....
<h3>Deep frying</h3>
<h2>Cooking with butter</h2>
... text of the section ...
</body>
```

Référence WCAG 2.0 : 1.3.1 ; 2.4.1 ; 2.4.6 ; 2.4.10

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 16 ; 23

Paragraphe

Les paragraphes servent à contenir des phrases. Généralement, un paragraphe contient au maximum trois ou quatre phrases corrélées logiquement, et donc différents paragraphes, c'est-à-dire différents tags <P> et </P>, apparaîtront dans un texte.

Pour revenir à la ligne, il faut utiliser le tag <br>, mais il est important que ce <br> soit inséré à l'intérieur du paragraphe, et donc entre les tags <P> ... </P>. L'utilisation des <br> pour séparer des paragraphes ou créer des espaces entre les paragraphes doit donc être considérée comme impropre.

La distance entre les différents paragraphes doit être gérée au moyen des CSS en donnant une valeur aux propriétés margin et padding du CSS du tag <P>.

Exemple de CSS pour écarter les paragraphes :

```
p {
margin-top: 1em;
margin-bottom: 1em;
}
```

Référence WCAG 2.0 : 1.3.1 ; 1.4.8

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 17 ; 23

Emphase

L'emphase est un aspect essentiel dans le texte, dans la mesure où elle focalise l'attention du lecteur sur des mots ou des sections auxquels l'auteur attribue une importance particulière.

Pour emphatiser le texte, en italiques ou en gras, il est conseillé de ne pas recourir au CSS, mais d'utiliser directement dans la structure (x)HTML les tags sémantiques <EM> et <STRONG>, pour générer respectivement l'italique et le gras.

Les tags <I> et <B> ont le même rendu visuel, mais, n'ayant pas de valeur sémantique, ils ne sont pas interprétés correctement par les outils d'assistance technologique et sont considérés comme des tags indésirables.

Exemple de la manière de mettre les textes en italique et en gras :

 Pour générer respectivement l' italique et le gras.

Référence WCAG 2.0 : 1.3.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 17

Listes

L'(x)HTML offre trois types différents de listes : ordonnées, non ordonnées et listes de définitions.

Les listes ordonnées, codifiées au moyen du tag d'ouverture <OL>, servent à décrire des éléments (<LI>) en séquence temporelle ou de priorité. Dans le texte, ces éléments sont énumérés en utilisant des lettres ou des numéros ordonnés.

Les listes non ordonnées, identifiées au moyen du tag d'ouverture <UL>, servent à regrouper les éléments (<LI>) où la numérotation ou l'ordre ne sont pas particulièrement importants. Leur utilisation convient à des listes simples, mais aussi à des menus de navigation, à des listes de nouvelles, etc.

Les listes de définition, identifiées au moyen du tag d'ouverture <DL>, sont des listes constituées

de termes (<DT>) avec la définition afférente (<DD>). Elles sont adaptées à structurer des sections du type nom-description ou produit-description, ou aussi à reproduire des dialogues du type interlocuteur-phrase.

Exemple de la manière de régler les différents types de listes :

```
<ol>
 <li>Mix eggs and milk in a bowl.</li>
 <li>Add salt and pepper.</li>
</ol>

<ul>
 <li>Milk</li>
 <li>Eggs</li>
 <li>Butter</li>
</ul>

<dl>
 <dt>blink</dt>
 <dd>turn on and off between .5 and 3 times per second
 </dd>
</dl>
```

Référence WCAG 2.0 : 1.3.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 17

Citations

Les citations doivent être marquées en utilisant les tags <Q> et <BLOCKQUOTE>, qui ne doivent pas être utilisés pour des effets esthétiques, mais pour fournir au navigateur et aux outils d'assistance technologique les informations nécessaires pour identifier le bloc de citation, en laissant au CSS les éventuels effets d'indentation des textes.

Le tag <BLOCKQUOTE> est utilisé pour identifier un bloc de texte de citation qui requiert une visibilité accrue, pour le séparer et le distinguer du corps du texte normal. Le code présent à l'intérieur de blockquote doit être compris dans un autre élément, comme .

Le tag <Q> est utilisé pour des citations brèves contenues à l'intérieur d'un paragraphe.

L'attribut CITE, qui reproduit l'URI du document source de la citation, est utilisable aussi bien par <BLOCKQUOTE> que par <Q>.

Les navigateurs visualisent généralement le <BLOCKQUOTE> comme un bloc de texte indenté, tandis que le texte présent dans le <Q> est reproduit à l'intérieur de guillemets ou dans des styles de citations différents dans le contexte de la langue utilisée.

Exemple de la manière d'organiser les citations :

<blockquote cite="http://www.mycom.com/tolkien/twotowers.html"> They went in single file, running like hounds on a strong scent, and an eager light was in their eyes. Nearly due west the broad swath of the marching Orcs tramped its ugly slot; the sweet grass of Rohan had been bruised and blackened as they passed. </ blockquote >

John said, <q>I saw Lucy at lunch, she told me <q>Mary wants you to get some ice cream on your way home.</q> I think I will get some at Ben and Jerry's, on Gloucester Road.</q>

Référence WCAG 2.0 : 1.3.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 17

g. Actualisation et redirection

Dans la section <HEAD> de l'(x)HTML, le méta-tag <META HTTP-EQUIV="REFRESH"> permet, en variant les attributs du tag, aussi bien la mise à jour automatique (refresh) d'une page web que la

possibilité de réadresser (redirect) la page vers une autre.

Il faut éviter les actualisations périodiques de page à travers le méta-tag <HTTP-EQUIV="REFRESH"> : un utilisateur particulièrement lent dans la lecture pourrait ne pas avoir le temps de lire les contenus. Il est possible de créer des pages qui s'actualisent automatiquement, mais il faut prévoir un mécanisme de blocage.

De même, il faut éviter l'utilisation du même méta-tag pour forcer le chargement d'une autre page. Préférer des mécanismes de réadressage côté serveur ou bien prévoir une page statique avec le lien permettant de passer à la nouvelle page.

Exemple de code à éviter :

- Ne pas utiliser de meta pour le rafraichissement au bout de x secondes :
  <meta http-equiv='refresh' content="300">
- Ne pas utiliser de meta pour le redirection au bout de x secondes de la page test.html :
  <meta http-equiv='refresh' content="300;url=test.html">

Référence WCAG 2.0 : 2.2.1 ; 2.2.2 ; 2.2.3 ; 2.2.4 ; 3.2.5

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 53

3. CSS (Cascading Style Sheets)

Comme nous l'avons déjà indiqué dans le paragraphe II.1, les CSS (Cascading Style Sheets) permettent de définir la présentation graphique (type de caractère, couleur, fond et espacement d'un texte) à appliquer aux documents (x)HTML.

Les instructions de style doivent être mises dans un fichier .css extérieur au fichier .html et non mélangées aux contenus (x)HTML, non seulement aux fins de l'accessibilité mais aussi pour un entretien plus simple du site.

Les navigateurs élaborent les CSS en exécutant les instructions en séquence (« en cascade ») :

- User agent CSS : feuille de style propriétaire du navigateur ;

- Author CSS : feuille de style créée par l'auteur de la page web ; cette feuille de style prévaut sur la précédente ;

- User CSS : il s'agit d'un type particulier de feuille de style, généralement créée par les utilisateurs ayant des exigences spécifiques, comme les malvoyants, qui ont la faculté de l'activer au moyen de personnalisation du navigateur ; cette feuille de style a la priorité absolue sur les précédentes.

Sur la base de ces considérations, dans la définition des propriétés du CSS, le développeur doit tenir compte du fait que la page web pourrait être visualisée avec un CSS défini par l'utilisateur (user CSS) : l'user CSS définit uniquement les tags (x)HTML relatifs aux contenus (titres, listes, etc.) et non les tags (div et span) relatifs au lay-out de la page (titre, menu principal, etc.).

Les règles pour composer les feuilles de style sont contenues dans un ensemble de lignes directrices déjà publiées par le W3C. Voici quelques suggestions de codification correcte du CSS utiles en vue de l'accessibilité :

- Suivre correctement les règles indiquées par le « box model » du CSS. Pour simplifier la largeur et la hauteur de chaque élément définissable comme box (par exemple les listes, les cellules de tableaux, les divisions, les paragraphes, etc.), il faut en indiquer les propriétés dans le CSS :

  - width (largeur)
  - height (hauteur)
  - padding (marge intérieure)
  - border (largeur du bord)

  En sachant que, dans le cas d'Internet Explorer 5.*/Win, les deux valeurs relatives à la marge intérieure et à la largeur du bord sont incluses dans la largeur et la hauteur fournies.

- Spécifier les propriétés de box selon l'ordre correct haut, droite, bas, gauche. Le CSS permet d'indiquer dans une instruction unique les valeurs relatives aux propriétés margin, padding ou border, mais il est nécessaire de rappeler que l'ordre d'interprétation est dans le sens horaire à partir du haut par rapport au box (haut, droit, bas, gauche – Top, Right, Bottom, Left).

- Spécifier les unités de mesure pour les valeurs numériques autres que zéro. Les règles du CSS veulent que, pour certaines propriétés, par exemple width, height et font-size, les unités de mesure soient clairement spécifiées, sauf dans le cas où la valeur est zéro.

- Éviter le hack en utilisant les commentaires conditionnels pour Internet Explorer dans la

section <HEAD> de l'(x)HTML. Exemple :


```
<!-- [if IE]>
<link rel="stylesheet" type="text/css" href="style_IE.css" media="screen">
<! [endif] -->
```

- Spécifier les styles relatifs aux liens. Le CSS doit définir le style des liens selon l'ordre suivant :
  - a:link{ }
  - a:visited{ }
  - a:hover{ }
  - a:active{ }
- Certaines informations nécessaires aux utilisateurs de lecrteurs d'écran, ne doivent pas être affichées sur l'interface. L'utilisation de l'instruction DISPLAY:NONE ne permettant pas de réaliser cela, il est recommandé d'utiliser les instructions suivantes :
  - position : absolute
  - top : -10000px
  - left : -10000px

  D'autres modes de codifier le texte caché impliquent que le lecteur d'écran ignore aussi ce texte :
- Dénommer correctement les classes et id du point de vue syntaxique.

  Les noms de classes et id peuvent contenir seulement des caractères alphanumériques [A-Za-z0-9] et le tiret (-) et ne peuvent pas commencer par un tiret ou un chiffre.
- Considérer que les propriétés CSS sont sensibles à la casse.

  Les valeurs des attributs (x)HTML class et id sont aussi sensibles à la casse utilisées dans la feuille de style. Par exemple id="HOME" identifie un objet différent de id="home".
- Définir les id de manière univoque.

  Il faut garder dans l'esprit que, dans un document (x)HTML, l'attribut id est utilisé pour identifier de façon univoque un élément. Au contraire, la classe peut être rappelée

plusieurs fois, dans le code (x)HTML, pour caractériser de la même façon des éléments différents.

- Codifier correctement les commentaires. Dans les CSS, les commentaires doivent être précédés de /* et suivis de */ : (exemple : /* Ceci est un commentaire */).
- Organiser le formatage et la position des éléments sur la page, par exemple :

  - marges : pour créer des espaces sur les quatre côtés du contenu d'un élément, on doit utiliser les propriétés « margin » (margin-top, margin-right, margin-bottom, margin-left) ou les propriétés « padding », au lieu d'ajouter des espaces ( ) ;

  - indentations: utiliser 'text-indent' ;

  - espacement entre lettres et/ou mots : utiliser letter-spacing et/ou word-spacing, ne pas ajouter des caractères d'espace pour espacer les lettres ;

  - images et couleurs de fond ;

  - polices, tailles et couleur des caractères ;

  - positionnement des différents éléments dans une image (mise en page), en respectant correctement les règles relatives aux éléments 'float'.

Référence WCAG 2.0 : 1.3.1 ; 1.3.2 ; 1.4.8 ; 4.1.1

Type de contrôle :

☒ Test automatique

☐ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.a et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 5

4. Script et autres objets de programmation

Comme nous l'avons déjà spécifié au paragraphe II.1, à l'intérieur des sites/applications, les actions se réalisent au moyen des fonctions JavaScript ou d'autres objets de programmation. Toutefois, toutes les technologies ne supportent pas ceux-ci. Dès lors, dans l'optique de l'amélioration progressive, il est nécessaire de prévoir des modalités alternatives à ces actions.

a. Alternatives aux scripts, applets ou autres objets de programmation

Toutes les informations significatives et les fonctionnalités activables par scripts, applets et autres objets de programmation doivent être disponibles également quand les pages sont visualisées avec des navigateurs et des technologies qui ne les supportent pas.

Pour ce motif, les informations et les fonctionnalités véhiculées au moyen de ces objets doivent être également disponibles dans des modalités alternatives.

Naturellement, l'alternative doit être fournie dans le cas où ces objets ont un contenu informatif : un script qui change la couleur d'un texte au passage de la souris seulement à des fins esthétiques n'a pas besoin de modalités alternatives ou d'être commenté.

Si des scripts de validation sont prévus côté client, ces contrôles doivent être dupliqués côté serveur dans la mesure où certains navigateurs ne supportent pas JavaScript.

Exemple :

- Si l'on utilise JavaScript, l'information analogue peut être fournie à l'intérieur du tag <NOSCRIPT>
- Si l'on utilise <OBJECT>, il faut insérer la description textuelle à l'intérieur du tag <OBJECT> ;
- Si l'on utilise <APPLET>, il faut fournir un équivalent textuel à l'attribut ALT".

Référence WCAG 2.0 : 4.1.2

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.a, III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 9 ; 42

Concernant les mises à jour automatiques et les réadressages, les mêmes recommandations que celles déjà fournies pour l' (x)HTML au paragraphe II.2.e sont applicables : tant que les programmes utilisateur ne permettront pas de les bloquer, il ne faut pas utiliser de code JavaScript pour mettre à jour ou réadresser automatiquement les pages. Il faut préférer des mécanismes côté serveurs ou, dans le cas du réadressage, prévoir une page statique avec le lien pour passer à la nouvelle page.

b. Indépendance du dispositif d'input

La génération de tout événement, quel qu'il soit (ouverture d'un menu déroulant, activation de connexion, confirmation d'un formulaire, etc.) doit être :

- indépendante du dispositif d'input utilisé par l'utilisateur (souris, clavier, émulateur de souris, etc.) ;
- contrôlée par l'utilisateur et jamais activée automatiquement par le système comme cela se produit, par exemple, avec certaines fonctions JavaScript qui permettent la visualisation d'une information au simple passage de la souris ou au changement de l'élément d'une liste.

Il ne faut donc jamais utiliser de scripts qui requièrent uniquement l'utilisation de la souris, et il faut avoir à l'esprit que :

- OnDblClick, associé au double clic de la souris sur un élément HTML sélectionné n'a pas d'équivalent sur le clavier et ne doit donc pas être utilisé ;
- les événements de la souris liés à des coordonnées ne doivent pas être utilisés ;
- les événements de la souris doivent être associés à des éléments analogues du clavier. Par exemple :
  - OnMouseUp avec OnKeyUp ;
  - OnMouseDown avec OnKeyDown ;
  - OnClick avec OnKeyPress ;
  - OnMouseOver et OnMouseOut n'ont pas d'équivalent avec le clavier et doivent donc être associés, respectivement, avec OnFocus et OnBlur ;
  - ne pas activer des événements avec Onchange et OnSelect, mais insérer toujours un bouton explicite de confirmation.

Exemple sur OnSelect :

- Code à éviter

```
<script type="text/javascript">

<!--
```

```
function goto_URL(object) {
window.location.href = object.options[object.selectedIndex].value; }
//-->
</script>
<form action="page.htm" onsubmit="return false;">
<p><label for="selectPage">Go to:</label>
<select name="selectName" onchange="goto_URL(this.form.selectName)">
<option value="">Select a page:</option>
<option value="page.htm">Page 1</option>
<option value="page.htm">Page 2</option>
<option value="page.htm">Page 3</option>
</select></p>
</form>
```


- Code correct

```
<form action="page.htm" onsubmit="return false;">
<p><label for="selectPage2">Go to:</label>
<select name="selectPage2" id="selectPage2">
<option value="">Select a page:</option>
<option value="page.htm">Page 1</option>
<option value="page.htm">Page 2</option>
<option value="page.htm">Page 3</option>
</select>
<input type="button" value="Go!" onclick="goto_URL(this.form.selectPage2);" />
</p>
</form>
```

Référence WCAG 2.0 : 2.1 ; 2.1.1 ; 2.1.2 ; 2.1.3

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.a et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 24 ; 34 ; 52

c. Fenêtres et pop-up

Éviter l'ouverture de pop-up et de nouvelles fenêtres du navigateur par rapport à celle où est visualisé le site/l'application.

S'il n'est pas possible d'éviter ces événements, il est nécessaire d'avertir explicitement l'utilisateur : l'ouverture de nouvelles fenêtres et pop-up met en grande difficulté les personnes handicapées de la vue et les utilisateurs non habitués aux technologies informatiques.

Si la DTD transitionnelle de l'HTML a été adoptée, l'ouverture de nouvelles fenêtres est possible avec l'attribut TARGET du tag <A>. Dans ce cas, l'information à l'utilisateur pour l'ouverture de la nouvelle fenêtre doit être donnée directement dans le texte du lien ou à travers un texte caché.

Si la DTD stricte a été adoptée dans le développement de la page, l'ouverture de nouvelles fenêtres n'est de toute façon pas consentie parce que cette DTD ne prévoit pas l'attribut TARGET du lien.

Si l'on ne peut renoncer à l'ouverture de nouvelles fenêtres et qu'on utilise la DTD stricte, il est possible d'ouvrir de nouvelles fenêtres en utilisant JavaScript. Dans ce cas, il est alors nécessaire de fournir quand même le lien « réel » vers la page et d'avertir l'utilisateur de l'ouverture de la nouvelle fenêtre.

Cette méthode offre une alternative pour les technologies qui ne supportent pas JavaScript :

- Dans le cas où JavaScript est activé ou supporté, une fenêtre ou un pop-up s'ouvrira ;
- Dans le cas où il est désactivé ou n'est pas supporté, les contenus seront visualisés dans la même fenêtre.

L'avis pour l'ouverture d'une nouvelle fenêtre est « ouvre une nouvelle fenêtre », sans utiliser de parenthèses afin d'alléger le plus possible l'écoute du texte des liens.

Pour les précautions à appliquer pour les textes cachés, voir le paragraphe II.3.

Exemple de la manière de régler l'ouverture de nouvelles fenêtres :

L'exemple qui suit illustre la manière de gérer l'ouverture d'une nouvelle fenêtre dans le cadre de la DTD Strict, tant en présence qu'en l'absence de Javascript, et en en avertissant préalablement l'utilisateur.

- Dans la page (x)HTML :
  <a href=" http://www.addresspage.html" class="new_windows" onClick="javascript(this)"
  onkeypress="javascript(this)">

- Dans le code Javascript

```
onload=function()
{
if(!document.getElementsByTagName) return;
l=document.getElementsByTagName("a");
for(i=0;i<l.length;i++)
{
if(l[i].className.indexOf("new_window")!=-1)
{
l[i].title="ouvre une nouvelle fenêtre";
l[i].onclick=function(){window.open(this.href);return(false)};
}
}
}
```

À noter : le lien entre la classe (new_windows) définie dans le tag <a> et le script.

L'avis d'ouverture d'une nouvelle fenêtre est donné également dans le titre du lien du script (ouvre une nouvelle fenêtre).

Référence WCAG 2.0 : 2.1 ; 2.1.1 ; 2.1.2 ; 2.1.3 ; 2.4.6

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 31

5. Lay-out des pages

a. Caractéristiques générales

Les pages d'un site/d'une application doivent être organisées en zones bien définies, chacune d'elles étant destinée à accueillir des typologies spécifiques de contenus.

De façon classique, une page web est organisée comme suit :

- Le titre est positionné en haut et contient, en plus du logo, les fonctions de services et de métanavigations (retour à la page d'accueil, accès au guide/à l'aide, au plan du site, aux contacts, au moteur de recherche, aux zones réservées, etc.) ;
- Le menu principal est positionné :
  - horizontalement, sous le titre de la page, ou
  - verticalement, à gauche pour les langues occidentales, à droite pour les langues lues de droite à gauche, comme l'arabe ;
- Les contenus principaux sont positionnés sur le corps central de la page ;
- Des informations et fonctions additionnelles ou d'approfondissement prennent parfois place dans la zone droite ou gauche, selon le sens de la lecture.

Le lay-out pour les pages doit être maintenu de manière consistante dans tout le site/application. Le lay-out de la page d'accueil peut être différent du lay-out des pages intérieures.

Il est utile de décrire par un titre codifié par <H1> la fonction des différents groupes de liens ou zones de la page. Ce H1 peut être rendu invisible sur la page grâce aux instructions de style, mais lisible par les lecteurs d'écran. Pour les précautions à appliquer pour les textes cachés, voir le paragraphe II.3.

La description des zones doit avoir un caractère fonctionnel et ne pas utiliser de références spatiales (par exemple, en-tête) ou des termes techniques (par exemple, menu de métanavigation) :

- Utiliser « menu de service » pour la zone qui contient des liens vers des services, telle que

plan, recherche, guide, contacts, etc., typiquement placés en tête de page ;

- Utiliser « menu principal » pour la zone qui regroupe le menu principal du site/application, typiquement placé à gauche ;
- la partie du contenu est décrite avec le titre approprié selon ce qui est suggéré au paragraphe II.6.a.

Attention à ne pas découper le contenu de la page en un nombre important de zones, chacune codifiée par un texte caché, car il pourrait n'y avoir aucun avantage pour l'utilisateur qui utilise le lecteur d'écran, vu la redondance des contenus.

Le codage du lay-out de page doit être fait, de préférence, par <DIV> et CSS. Même si elle est plus lourde quant à la conformité de visualisation avec les différents navigateurs, cette solution garantit une compatibilité accrue et de meilleures possibilités d'entretien, mais surtout la nette séparation entre structure et présentation, qui a été évoquée au paragraphe II.1 et qui est nécessaire pour :

- le respect de la DTD de type Strict ;
- la lisibilité correcte si le CSS est désactivé, comme le font souvent les malvoyants ;
- la possibilité d'intervenir globalement sur le graphisme du site/de l'application en travaillant sur un fichier unique ;
- la garantie de l'uniformité de toutes les pages (consistance).

Exemple de la manière d'organiser le layout de page avec les CSS :

```
<head>
 <title>Stock Market Up Today</title>
 </head>
 <body>
 <!-- left nav -->
 <div class="left-nav">
 <h1>menu</h1>
 <!-- content here -->
 </div>
```

```
<!-- main contents -->

<div class="main">

<h1>Stock Market up today</h1>

<!-- article text here -->

</div>

<!-- right panel -->

<div class="link">

<h2>Related links</h2>

<!-- content here -->

</div>

</body>
```


Référence WCAG 2.0 : 1.3.1 ; 1.3.2 ; 2.4.6 ; 2.4.10

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 6 ; 10 ; 16

Il est possible de définir le lay-out de la page également au moyen de tableaux. Même si elle est très utilisée, cette pratique n'est pas conseillée, d'une part parce qu'il s'agit d'un usage impropre du tag <TABLE> et d'autre part parce qu'il présente des inconvénients :

- il est très difficile de réorganiser les zones de la page de manière différente ;
- les lecteurs d'écran tendent à les interpréter comme des tableaux de données (ils lisent, dans de nombreuses situations, le nombre de lignes et de colonnes) ;
- l'utilisateur qui emploie le lecteur d'écran est obligé de sauter ou d'écouter intégralement de longues listes de liens et de contenus secondaires avant de parvenir aux contenus principaux de la page.

Au cas où l'on choisit quand même une organisation basée sur les tableaux, il est nécessaire :

- d'utiliser seulement les tags <TABLE>, <TR> et <TD> et les fermetures afférentes ;
- de ne pas utiliser les tags <CAPTION>, <TH>, <THEAD>, < TBODY> et <TFOOT> parce que ces tags sont typiques des tableaux de données et, exécutés par les lecteurs d'écran dans des contextes de lay-out, ils rendent les contenus proposés incompréhensibles ;
- D'éviter que le lecteur d'écran interprète les tableaux comme des tableaux de données et en lise les nombres de lignes et de colonnes ;
- De ne pas loger les tableaux les uns dans les autres, mais d'utiliser un ou plusieurs tableaux pour chaque élément de la page (par exemple, un tableau pour l'en-tête, un tableau pour le corps central, un tableau pour le pied de page, etc.) ;
- De gérer les marges et les espaces avec les CSS ou <DIV> et non avec des éléments graphiques, des images de séparation ou des imbrications ;
- De prévoir des sauts de navigation (skip link) de façon à ce que l'utilisateur puisse sauter les menus et aller directement aux contenus principaux dans le corps central de la page (cf. paragraphe II.7.e).

Exemple de la manière d'organiser le layout de page avec tableaux et CSS :

```
<head>
 <title>Stock Market Up Today</title>
 </head>
 <body>
 <!-- left nav -->
<table>
<tr>
<td class="left-nav">
 <h1>menu</h1>
 <!-- content here -->
</td>
<!-- main contents -->
<td class="main">
```

```
<h1>Stock Market up today</h1>

<!-- article text here -->

</td>

<!-- right panel -->

<td class="link">

<h2>Related links</h2>

<!-- content here -->

</td>

</tr>

</table>

</body>
```


Référence WCAG 2.0 : 1.3.1 ; 1.3.2 ; 2.4.6 ; 2.4.10

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 11

b. Taille relative du lay-out

Utiliser le lay-out avec <DIV> et CSS ou l'utiliser avec des tableaux, dans ces deux cas, les différentes zones qui composent la page doivent être définies en pourcentage et jamais en valeur absolue. Ceci donne la possibilité aux navigateurs de procéder à l'adaptation des contenus en rapport avec les dimensions de l'interface utilisée par l'utilisateur sans superposition des objets ou perte d'informations.

Il faut prendre en considération le fait que les malvoyants travaillent avec des résolutions très basses, et donc la barre de défilement horizontal :

- ne doit pas apparaître à des résolutions supérieures à 600 px de largeur ;
- devrait apparaître de préférence à des résolutions comprises entre 480 px et 600 px de

largeur ;

- peut apparaître à des résolutions inférieures à 480 px de largeur.

Pour cette raison, comme cela est d'ailleurs indiqué au paragraphe II.10.a, il convient de faire attention au graphisme (par exemple, les grandes images placées dans les en-têtes des pages), qui empêche de modifier les dimensions de la page et entraîne l'apparition des barres de défilement horizontal.

Il faut aussi faire attention aux pages qui doivent être imprimées : dans certaines situations, il peut y avoir des coupures aux parties du contenu. La largeur d'une feuille A4 correspond en effet à environ 525 px.

Exemple de la manière de rendre un layout adaptable à la résolution de l'écran :

- Dans la page (x)HTML :

```
<div id="wrap">

<div id="main"></div>

<div id="sidebar"></div>

</div>
```

- Dans le code CSS :

```
#main {

float:left;

width:60%;

}

#sidebar {

float:right;

width:40%;

}
```

Référence WCAG 2.0 : 1.3.1 ; 1.4.4 ; 1.4.5 ; 1.4.8

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.a et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 12

c. Compatibilité avec les différents navigateurs

Les pages doivent être réalisées de manière à être compatibles avec les navigateurs les plus utilisés dans les différents systèmes d'exploitation. On entend par compatibilité que :

- les contenus et fonctionnalités soient les mêmes avec tous les navigateurs graphiques ;
- la présentation graphique soit semblable dans les différents navigateurs graphiques.

Les navigateurs à prendre en considération sont les suivants :

- Windows :
  - Internet Explorer
  - Opera
  - Firefox
  - Google Chrome
- Mac OS X :
  - Firefox
  - Safari
- Linux :
  - Konqueror
  - Firefox

La compatibilité doit être garantie également avec les versions de navigateurs en usage au moins deux ans avant la publication de la page web.

Il convient aussi de tenir compte de Lynx, un navigateur textuel qui visualise seulement les contenus textuels des pages, ne supporte pas la souris, les images (montrant donc le texte alternatif), les enregistrements vidéo et audio, les animations, les plug-ins, Java et JavaScript. Il est typiquement utilisé par les aveugles, avec l'aide d'un lecteur d'écran.

Référence WCAG 2.0 : 1.3.1 ; 4.1.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 7 ; 8 ; 14

6. Présentation des contenus

a. Organisation des titres

Utiliser les tags (x)HTML adéquats, indiqués au paragraphe II.2.d pour structurer le texte à l'intérieur de la page : <H1> pour le titre principal, <H2> pour les titres de deuxième niveau, <H3> pour les titres de troisième niveau, et ainsi de suite jusqu'à <H6>. Ne pas utiliser ces tags pour obtenir des effets de formatage.

Ne pas utiliser trop de niveaux d'indentation à l'intérieur d'une seule page.

Le titre principal (<H1>) doit être significatif, de manière à aider l'utilisateur à comprendre à quoi se réfèrent les données et les informations contenues dans la page et/ou quelle activité doit être effectuée.

Les titres ne doivent pas dépasser la longueur maximale de 80 caractères (y compris les espaces) : le texte excédentaire n'est pas lu par le lecteur d'écran et l'utilisateur perd l'information sur la hiérarchie entre les différents titres.

Le fichier (x)HTML doit aussi avoir un titre significatif inséré dans le tag <TITLE> placé entre le début et la fin de <HEAD>, de manière à ce que l'utilisateur puisse reconnaître la page sans équivoque lorsqu'il en insère l'adresse dans la liste des favoris. Le standard pour le <TITRE> des fichiers html est :

```
<TITLE>Nom site - Titre page visualisée</TITLE>
```

Le titre qui identifie les contenus principaux et le titre du tag <TITLE> doivent être cohérents entre eux, comme indiqué au paragraphe II.7.b.

Il faut faire attention à la longueur du TITLE : pour ne pas avoir le texte tronqué, la description dans le <TITLE> ne doit pas dépasser les 70 caractères (espaces compris).

Dans le <TITLE>, comme dans les autres parties textuelles du site/de l'application, il ne faut pas utiliser les signes supérieur (>) et inférieur (<) car ils sont interprétés de façon littérale par les lecteurs d'écran. Il faut préférer un tiret, les deux points ou les barrettes verticales.

N'utiliser le caractère majuscule que pour la première lettre du premier mot des titres des pages (par exemple, « Paramétrage de la recherche » au lieu de « Paramétrage de la Recherche »), sauf s'il s'agit d'un acronyme ou d'un nom propre.

Ne pas utiliser les verbes à l'infinitif (par exemple « Paramétrage de la recherche » plutôt que « Paramétrer la recherche »).

Exemple de la manière d'organiser le titre HTML et les titres <H> :

```
<head>
 <title> Ministère des Finances - Services</title>
</head>
 <body>
 ……
 <!-- main contents -->
 <div class="main">
 <h1> Services du Ministère</h1>
 <!-- text here -->
 </div>
 …….
</body>
```

Référence WCAG 2.0 : 1.3.1 ; 2.4.1 ; 2.4.2 ; 2.4.6 ; 2.4.10

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 15 ; 16

b. Style rédactionnel

Présenter les textes de la façon la plus simple possible, structurés en phrases brèves, en éliminant tout ce qui est superflu. Si nécessaire, dans le cas de documents très longs, présenter au début une synthèse significative en renvoyant au texte complet via un lien.

Utiliser toujours les mêmes mots pour indiquer les mêmes actions ou les mêmes concepts : si l'on a choisi de parler de « zones du site », ne pas les changer en « sections du site ».

Ne pas utiliser des mots étrangers ni un langage que des utilisateurs non experts en informatique ou qui ne comprennent pas les langues étrangères ne peuvent pas comprendre. Chercher le plus possible à ne pas les utiliser dans la mesure où la plupart des lecteurs d'écran les lisent tels quels (par exemple, « download » est lu « do double vé load », « file » est lu comme le mot français « file », au sens de « queue »).

Il faut toujours spécifier la langue prédominante de la page ou d'une portion de texte à travers l'attribut LANG, comme indiqué au paragraphe II.2.c.

Éviter autant que possible les acronymes et les abréviations parce que les lecteurs d'écran ont des difficultés à les interpréter. S'il est nécessaire de recourir aux acronymes et aux abréviations, éviter de les spécifier au moyen de ACRONYM (exemple : <ACRONYM TITLE= "Bons ordinaires du trésor">BOT</ACRONYM>) et ABBR (exemple : <ABBR TITRE="Paragraphe">Par.</ABBR>) mais préférer la mention étendue des mots ou la spécification entre parenthèses du sens de l'acronyme, au profit de l'ensemble des utilisateurs.

Ne pas présenter de texte clignotant ou en mouvement dont les fréquences d'intermittence peuvent provoquer des troubles d'épilepsie photosensible, des troubles de la concentration ou qui peuvent causer le mauvais fonctionnement des outils d'assistance technologique. La fréquence de clignotement qui peut créer des troubles d'épilepsie est celle comprise entre 4 et 59 Hz. Lorsque des informations nécessitent l'utilisation de ces animations, il faut prévenir l'utilisateur du risque possible avant de les présenter et prévoir des méthodes qui permettent de les éviter.

Si l'on veut mettre en évidence un contenu, eviter de le souligner afin de ne pas le confondre avec un lien, utiliser plutôt le gras. Rappelons que les caractères gras doivent être codifiées avec <STRONG> et non avec <B> (bold), comme indiqué au paragraphe II.2.d L'utilisation du gras doit être toutefois modérée (par exemple, ne pas l'utiliser pour les labels des contrôles) De plus le gras ne doit pas être associée à une diminution de la taille des caractères.

Ne pas utiliser les italiques, car elles sont plus difficiles à lire à l'écran. Quand il faut les utiliser,

elles doivent être codifiées avec <EM> et non avec <I> (italic), comme indiqué au paragraphe II.2.d.

Ne pas écrire des parties entières de texte en majuscule parce qu'elles sont plus difficiles à lire par rapport aux minuscules ; les majuscules peuvent être utilisées pour des phrases qui doivent attirer l'attention, du moment qu'elles sont très brèves.

S'il faut insérer du texte caché au bénéfice des utilisateurs qui emploient les lecteurs d'écran, consulter les indications du paragraphe II.3.

Référence WCAG 2.0 : 2.3.1 ; 2.3.2 ; 3.1.1 ; 3.1.4 ; 3.1.5 ; 3.2.4 ; 4.1.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : difficile

Objet de la check-list d'accessibilité : 21 ; 22

c. Style graphique

Pour le paramétrage graphique des contenus et leur disposition sur la page, utiliser les CSS et les organiser de façon à ce qu'ils puissent être lus également lorsqu'ils sont désactivés ou qu'ils ne sont pas supportés.

Pour définir la dimension du texte, y compris les étiquettes des champs d'input, utiliser la taille relative et non la taille fixe (en pixels).

Comme type de police de caractères, préférer celles sans serif (arial, verdana, tahoma, helvetica, etc), qui sont plus lisibles à l'écran et utiliser toujours les polices du système. Pour les applications, il est préférable d'utiliser arial, tandis que verdana sera choisi pour les sites uniquement de texte.

Pour toutes les polices de caractères, y compris les étiquettes et les textes des contrôles et des champs d'édition, utiliser la taille relative et non la taille fixe (en pixels), de façon à ce que les contenus puissent s'adapter à l'interface employée par l'utilisateur sans superposition des objets ou perte d'informations de nature à rendre incompréhensible le contenu, même en cas de modification des dimensions, d'agrandissement ou de réduction de la zone de visualisation et/ou des caractères par rapport aux valeurs par défaut. Pour la dimension standard des polices, on peut utiliser les paramètres suivants, correspondant à 13 px, 10 pt, font-size 2 :

- 80%, ou
- 0.8 em, ou
- small.

Le texte doit toujours être aligné à gauche ou à droite selon le sens de lecture de la langue. Ne pas utiliser, sauf pour des cas particuliers (par exemple, le titre de la page), le texte centré, justifié ou aligné du côté opposé à celui de la direction de lecture.

Comme indiqué au paragraphe II.10 :

- ne pas utiliser la couleur comme forme unique de communication ;
- utiliser toujours un contraste fort en termes de luminosité et de couleur entre le fond et le texte, de manière à rendre facile la distinction par des personnes ayant des problèmes de perception de la couleur ;
- éviter de présenter des textes sous forme d'images parce que les images ne peuvent pas être agrandies de façon adéquate.

Exemple d'information codifiée NON UNIQUEMENT avec la couleur :

L'exemple qui suit illustre comment rendre explicite, au moyen de l'astérisque, l'information fournie par la couleur.

- Dans la page (x)HTML :

```
<label for="email" class="required">E-mail *:</label>

<input id="email" type="text" size="25" value=""/>


<label for="lastname">Last name:</label>

<input id="lastname" type="text" size="25" value=""/>
```

- Dans le code CSS:

```
.required {

color: red;

}
```

Exemple de la manière de rendre le texte relatif à travers le CSS :

```
body { font-size: 100%; }
```

Référence WCAG 2.0 : 1.3.1 ; 1.4.1 ; 1.4.4 ; 1.4.8

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.a et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 10 ; 13 ; 20

d. Numéros et caractères spéciaux

Dans leur paramétrage par défaut, les lecteurs d'écran interprètent toutes les séquences de chiffres comme des nombres entiers, sauf s'ils commencent par 0. Il peut donc y avoir des difficultés de compréhension de la part des utilisateurs dans le cas de numéros de téléphone ou d'autres caractères spéciaux.

Il peut y avoir des problèmes de compréhension pour les numéros de téléphone en tunisie (par exemple, le numéro 71710170 sera lu de la manière suivante: soixante et onze millions, sept cent dix mille, cent soixante dix) ce qui rend la compréhension et la mémorisation difficiles.

Pour faire en sorte que les numéros de téléphone qui ne commencent pas par 0 soient facilement lisibles et mémorisables, il faut adopter les règles suivantes pour :

- numéros tunisien : deux chiffres pour l'indicatif régionale, puis groupes de trois chiffres séparés par un tiret ou deux espaces (par exemple, 71-710-170 ou 71 710 170) ;
- numéros tunisien avec indicatif international : deux zero ou le signe +, puis trois chiffres du préfixe international, suivis des deux chiffres de l'indicatif régionnal et enfin groupes de trois chiffres séparés par un tiret ou deux espaces. (par exemple, 00 216 71 710 170 / 00-216-71-710-170 / +216 71 710 170 / +216-71-710-170

Le même numéro de l'exemple précédent, s'il est écrit « 71 710 170 », sera lu « Soixante et onze, sept cent dix, cent soixante dix », donc de manière plus compréhensible.

Pour faire lire correctement les numéros ordinaux (premier, deuxième, troisième, etc.) il faut les écrire 1er, 2eme, 3eme, etc. ou en entier (premier, deuxième, troisième, etc.).

Les caractères romains sont interprétés de manière irrégulière et souvent comme du texte normal (I=i, II=deux, VI=vi, VII=sept). Il est donc déconseillé de les utiliser.

Pour lire « n. » comme « numéro » (par exemple n. 32 = numéro 32), il faut écrire « n° » (par exemple n°32).

Référence WCAG 2.0 : 3.1.2 ; 3.1.3 ; 4.1.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.b selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : chapitre «Observations additionnelles»

7. Navigation et liens

a. Caractéristiques générales

Il faut organiser la navigation sur la base des caractéristiques des actions que les utilisateurs devront effectuer en interaction avec le site/l'application.

Compte tenu des limites des pages web (visualisation d'une page à la fois), on tend souvent à reproduire une logique de navigation qui donne à l'utilisateur l'opportunité de faire seulement un choix par page. Un bon regroupement des liens, disponibles dans des zones dédiées de l'écran permet d'éviter, du moins en partie, ces limitations en rendant l'interaction davantage flexible.

Il y a lieu de réduire le plus possible le nombre de pages à consulter pour arriver à l'objectif de l'activité. L'utilisateur devrait pouvoir accéder à l'information qui l'intéresse en naviguant au maximum à travers trois pages.

Permettre le retour à la page d'accueil et l'accès aux fonctionnalités du service (aide en ligne, moteur de recherche, plan, fonction de déconnexion, etc.) à partir de chaque page du site/de l'application.

Créer un ordre logique de navigation (par la touche de tabulation du clavier) entre les liens en utilisant, si nécessaire, l'attribut TABINDEX.

Éviter que l'utilisateur se fie aux instruments de navigation du navigateur : mettre toutes les options de navigation sur la page.

Maintenir dans l'ensemble du site/de l'application la consistance des mécanismes et le style de navigation.

Éviter les « Voie sans issue », c'est-à-dire les pages sans options de navigation.

Si le site est très complexe, insérer un moteur de recherche d'utilisation simple mais efficace.

Éviter l'utilisation de l'attribut ACCESSKEY car celle-ci peut alourdir sensiblement la lecture des contenus par les lecteurs d'écran. ACCESSKEY peut être utilisé uniquement dans les cas où il y a des utilisateurs quotidiens et pour des fonctions répétitives, comme cela peut être le cas dans l'utilisation d'application ou de sites intranet. Dans ces cas, dans le choix de la lettre à utiliser comme access key, il faut faire attention de ne pas créer d'interférence avec les raccourcis clavier du navigateur, du système d'exploitation et des outils d'assistance technologique.

Pour le codage des liens du menu, le tag prévu pour les listes non ordonnées (<UL LI>) peut être utilisé à condition que le lecteur d'écran ne lise pas ces listes avec une redondance excessive. L'abondance dans la lecture crée en effet d'importantes difficultés de compréhension quand :

- les objets de la liste sont composés de peu de mots ;
- l'objet est visé par d'autres tags (est aussi un lien) ;
- il y a des listes à l'intérieur d'autres listes.

Comme indiqué au paragraphe II.5.a, il est utile de décrire avec un titre codifié en utilisant le <H1> et la fonction des différents groupes de liens à travers un texte éventuellement caché ou même invisible sur la page grâce aux instructions de style mais lisible par les lecteurs d'écran.

Éviter l'ouverture de pop-up et de nouvelles fenêtres du navigateur par rapport à celle où le site/l'application est visualisé. Si ce n'est pas possible, voir les indications fournies au paragraphe II.4.c.

Si un plan graphique est utilisé pour la navigation ou la présentation des liens, celui-ci doit être défini côté client et non côté serveur. Afin de garantir l'accessibilité des plans, se référer aux indications fournies au paragraphe II.10.e.

b. Reconnaissance du contexte

Mettre absolument en évidence le contexte dans lequel se trouve l'utilisateur à la suite des sélections effectuées. Il convient dès lors de toujours présenter et rendre évidents :

- le tag <TITLE> du fichier (x)HTML et le titre <H1> de la page ;
- le lien sélectionné ;
- si possible, la position de la page par rapport à la structure globale du site/de l'application.

Pour ce dernier aspect, on peut utiliser les breadcrumb, ou « miettes de pain », ou le fil d'Ariane, communément repris dans les pages web sous la mention « vous êtes dans : ».

Le fil d'Ariane met en évidence la position de la page à l'intérieur du site/de l'application, en

montrant le parcours effectué à partir de la page d'accueil, et il se présente comme un accessoire utile pour la navigation. Il doit respecter la position de la page à l'intérieur du site/de l'application, et non pas nécessairement les étapes de la navigation effectuée (par exemple, si un lien porte à une zone interne du site différente de celle où l'on se trouve, le parcours doit mentionner la nouvelle zone interne).

Pour séparer les mots dans le fil d'Ariane, il ne faut pas utiliser les signes supérieur (>) et inférieur (<) parce qu'ils sont lus de manière inappropriée par certains lecteurs d'écran. Par exemple, la série « Page d'accueil > page intermédiaire > page visualisée » est lue de cette manière par certains lecteurs : « Page d'accueil fin de parenthèse pointue page intermédiaire fin de parenthèse pointue page visualisée ». Il faut préférer un tiret ou les barrettes. La solution conseillée pour le fil d'Ariane est la suivante :

Vous êtes dans : page d'accueil – nom page intermédiaire – nom page visualisée

où :

- les mots précédant la page visualisée sont des liens : ils fonctionnent comme support pour la navigation en permettant le retour aux pages précédentes ;
- le dernier mot, en quelque sorte mis en évidence, représente le titre de la page visualisée. Ce mot doit être répété comme en-tête de la page en utilisant <H1>.

Maintenir la correspondance entre le texte utilisé pour les liens et les titres des pages associées.

Le lien de la page visualisée doit être désactivé, par exemple, si l'on se trouve sur la page d'accueil, le lien vers la page d'accueil ne doit pas être actif et mis en évidence d'une manière ou d'une autre.

Exemple de la manière d'organiser le fil d'Ariane :

```
<head>
 <title> Ministère des Finances - Services</title>
</head>
 <body>
 ……
```

```
<!-- main contents -->

<p> Vous vous trouvez dans : <a href="index.htm">Home</a> - Services</p>

<div class="main">

<h1>Services du Ministère</h1>

<!-- text here -->

</div>

……..

</body>
```

Référence WCAG 2.0 : 2.4.7 ; 2.4.8 ; 2.4.10

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 6 ; 15

c. Activation et caractère significatif des liens

Prévoir la compréhension des contenus des pages de manière à ce que l'utilisateur puisse clairement comprendre ce qu'il trouvera dans les différentes pages avant même d'y accéder.

Les liens doivent être décrits de manière synthétique et significative, même s'ils sont lus hors de leur contexte, comme dans la modalité d'utilisation par les aveugles au moyen du lecteur d'écran, qui peut extraire de la page une liste des liens présents. Pour atteindre ces objectifs, il peut parfois être nécessaire d'adapter le texte de la phrase qui contient les liens.

Dans tous les cas, les liens ne doivent pas dépasser la longueur maximale de 110 caractères (espaces compris) ; le texte excédant est « tronqué » par le lecteur d'écran.

Pour clarifier la signification du lien, on peut utiliser l'attribut TITLE du tag <A>, en tenant toutefois compte du fait que :

- Certains outils d'assistance technologique ne parviennent pas à le lire ;

- D'autres outils d'assistance technologique permettront, selon les cas, de visualiser et/ou lire :
    - Soit le texte des liens ;
    - Soit le texte spécifié à travers le TITLE ;
    - Soit les deux, ce qui crée de la confusion lorsque le texte du lien et celui du TITLE sont différents l'un de l'autre.

De plus, l'attribut TITLE n'est pas supporté par tous les navigateurs et ne peut donc pas être considéré comme un instrument principal pour communiquer des informations importantes : il peut être utile pour augmenter l'expressivité du lien, mais ne doit pas être indispensable à sa compréhension. Il n'y a pas de motif d'ajouter le TITLE aux liens en spécifiant le texte même du lien. Le texte contenu dans le TITLE ne doit pas dépasser les 60 caractères.

Il faut prêter une grande attention à la consistance dans la dénomination des liens. Si, sur la même page, il y a deux liens qui conduisent à la même page (par exemple, un lien sur le menu et un lien dans la page de contenu), le texte et l'éventuel TITLE doivent être identiques pour les deux liens.

De façon analogue, il ne peut y avoir de liens avec la même dénomination qui conduisent à des pages différentes.

Dans tous les cas, éviter de dupliquer les liens si ce n'est pas strictement nécessaire.

Exemple de la manière d'organiser le skip link :

Le congrès Informatique et personnes handicapées s'est tenu à Tunis... continuer >>
Le congrès Informatique et personnes handicapées s'est tenu à Tunis.
Le congrès Informatique et personnes handicapées s'est tenu à Tunis.

Dans ces trois cas, le premier lien n'a pas de sens s'il est lu hors du contexte de la phrase ; le deuxième lien pourrait être trop abondant ; le troisième lien véhicule l'information de façon appropriée même s'il est lu séparément du reste de la phrase.

« Pour vous inscrire, cliquez ici » a du sens s'il est lu dans la phrase entière, mais perd sa signification si on lit ou si on écoute seulement « cliquez ici ». Le même pour Suivant, Approfondissement, etc.

Dans les liens du courrier électronique, il faut éviter d'écrire uniquement le nom du destinataire du message électronique (par exemple, Flen Ben Foulen, Service d'assistance) car on pourrait être amené à penser qu'il s'agisse d'une page reliée et non d'un système pour envoyer un message. Il vaut mieux écrire : Envoyer un courriel au service d'assistance ou expliciter directement l'adresse électronique (par exemple : assistance@administration.tn).

Lorsque c'est nécessaire, utiliser les liens pour le retour à la page précédente et l'avance à la page suivante, éviter d'utiliser des termes génériques comme « Précédent » et « Suivant » car ils ne sont pas significatifs s'ils sont lus hors de leur contexte et parce ce qu'ils sont ambigus pour tous les utilisateurs. S'il est nécessaire de recourir à ces liens, il est préférable d'utiliser des textes significatifs. Par exemple, si l'utilisateur se trouve dans la page des résultats de la

recherche, pour revenir à la page de paramétrage de la recherche, il vaut mieux utiliser le lien « paramétrage de la recherche » à la place de « page précédente » puisque l'utilisateur peut avoir oublié quelle était la page précédente.

Ne pas utiliser les URL comme texte des liens, surtout si elles sont complexes. Par exemple, dans le cas de la phrase « Le logiciel peut être téléchargé à la page suivante du site de Microsoft : http://www.microsoft.com/downloads/include/en-us/RelatedSites.asp », le lecteur d'écran lira l'URL, qui apparaîtra incompréhensible, et les utilisateurs recourant à des systèmes d'input vocal auront de grandes difficultés à donner la commande vocalement. Il vaut mieux présenter le lien avec un texte : « Le logiciel peut être téléchargé dans la section du site Microsoft consacré au téléchargement »..

Si, pour quelque motif que ce soit, on veut indiquer explicitement si le lien est relié à des pages internes ou à des pages externes au site, il faut le rendre avec le choix du texte du lien et non avec des codages graphiques particuliers (par exemple : Forum pour le lien intérieur, et Forum du site du gouvernement pour les liens extérieurs).

Il faut éviter d'utiliser des acronymes, abréviations et mots étrangers comme texte des liens, car il pourrait y avoir des ambiguïtés sur la façon correcte de prononcer les liens pour les utilisateurs qui utilisent des systèmes d'input vocal.

Référence WCAG 2.0 : 2.4.4 ; 2.4.9

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 25 ; 26

d. Caractère reconnaissable des liens

Tous les liens doivent être sélectionnables et activables au moyen du clavier, des technologies d'émulation du clavier ou d'autres systèmes de pointage différents de la souris.

Les liens doivent être rapidement distingués des autres éléments non sélectionnables à travers la seule vision directe, sans devoir vérifier leur caractère sélectionnable ou non avec le pointeur de la souris.

Le meilleur moyen d'assurer ce caractère reconnaissable est garanti par l'utilisation du style bleu souligné, pour des motifs conventionnels, en exploitant le style standard du tag <A>. Les autres alternatives possibles du style présentent les problèmes suivants :

- la couleur perd son efficacité en cas de défauts de la perception de la couleur ;
- le gras ne peut pas être utilisé pour mettre en évidence des parties du texte normal, car il pourrait y avoir ambiguïté.

Le lien déjà visité doit être rendu évident en utilisant, par exemple, le violet souligné

conventionnel.

Il peut être utile de spécifier également le hover du lien, surtout si l'effet est visible quand le focus se trouve sur le lien en utilisant le clavier et non uniquement au passage de la souris.

À l'intérieur d'un même site/application, les liens devraient être présentés toujours dans le même style graphique. Jusqu'à trois styles différents au maximum sont tolérés (par exemple un pour le menu principal, un autre pour le menu de service et un pour les liens dans la partie du contenu), du moment que leur caractère sélectionnable soit visible « à l'œil nu » par rapport aux autres éléments de la page.

Préférer les liens textuels par rapport aux liens graphiques, c'est-à-dire ceux réalisés au moyen d'images, car les malvoyants qui utilisent un agrandisseur éprouvent des difficultés à reconnaître une image agrandie et ne réussissent pas à lire de façon adéquate les ALT.

S'il s'avère nécessaire d'utiliser les liens graphiques, il est important :

- de rendre évident le caractère sélectionnable de l'élément ;
- d'ajouter toujours l'attribut ALT du tag <IMG> :

  - l'ALT devrait répéter le même texte que l'image pour favoriser les utilisateurs qui interagissent avec des systèmes d'input vocal ;

  - il n'est pas nécessaire de spécifier le mot « lien » parce que cette mise en évidence est donnée par le pointeur de la souris et par le lecteur d'écran pour les aveugles.

Il faut également avoir à l'esprit que les liens graphiques très petits (petites icônes, flèches et en général les puces) sont difficiles à sélectionner pour les handicapés moteurs.

Lorsque les liens sont précédés de puces ou équivalents, et que l'on veut mettre le renvoi également sur la puce, il faut implémenter un lien unique en utilisant de préférence les instructions CSS.

Référence WCAG 2.0 : 1.4.1 ; 3.2.4

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

e. Skip link

Chaque fois que l'utilisateur sélectionne un lien, les lecteurs d'écran commencent en fait à lire la page depuis le début. En fonction de la structure de la page, il peut arriver que des séquences répétitives de contenus ou des liens communs à toutes les pages soient relus chaque fois. Dans ces cas, il est toujours nécessaire de fournir des mécanismes pour sauter des groupes de liens ou des contenus répétitifs et amener l'utilisateur directement à la partie des contenus principaux de la page.

Le saut vers la partie du contenu principale (H1, dans le corps principal de la page) est possible à travers un lien à placer au début de la page, avant tout autre contenu. Ce lien doit être rendu invisible – mais lisible par les lecteurs d'écran – en utilisant toutes les instructions CSS suivantes, déjà indiquée au paragraphe 2.3 :

- position : absolute
- top : -10000px
- left : -10000px

D'autres manières de coder le texte caché, y compris l'instruction DISPLAY:NONE, ont pour effet que, même le lecteur d'écran ignore ce texte.

Les textes des skip link doivent être très directs et brefs. Par exemple, « contenus » pour le skip link qui renvoie directement aux contenus ; « menu principal » pour le lien qui renvoie directement au menu principal.

Exemple:

- Dans la page (x)HTML:

```
<body>

<div class="skiplink"><a href="#contenu">contenus</a> </div>

……

<!-- main contents -->
```

```
<p> Vous vous trouvez dans :  <a href="index.htm">Home</a> - Services </p>

<div class="main">

<a id="contenu" name="contenu"></a>

<h1>Services du Ministère </h1>

<!-- text here -->

</div>

……

</body>
```

- Dans le code CSS:

```
.skiplink {

position: absolute;

top: -10000px;

left: -10000px;

}
```


Référence WCAG 2.0 : 2.4.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 29

f. Espaces entre les liens

Les liens, regroupés dans le sens horizontal et vertical, doivent être opportunément espacés les uns des autres de manière à permettre aux handicapés moteurs d'atteindre aisément une cible.

Dans le cas de liens juxtaposés verticaux, il est nécessaire de paramétrer un espace interligne adéquat.

Dans le cas de liens juxtaposés horizontalement, une manière de bien rendre cette séparation consiste à insérer un tiret ou des barrettes verticales comme dans l'exemple suivant :

Exemple pour liens adjacents horizontalement :

Une manière de bien rendre cette séparation consiste à insérer un tiret ou des barrettes verticales comme dans l'exemple ci-après : Entrée de menu 1 | Entrée de menu 2

Référence WCAG 2.0 : 1.4.8

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.a selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 30

8. Formulaires

a. Caractéristiques générales

Organiser les formulaires en tenant compte des activités des utilisateurs. Prévoir le nombre, la fonction, l'ordre de présentation et l'organisation interne des formulaires en rapport avec le flux de travail de l'utilisateur et les informations qu'il traite aux différentes étapes de l'activité.

Utiliser une unité de travail appropriée. Il est opportun d'organiser les contenus des formulaires de façon à ce que les utilisateurs exécutent à travers ceux-ci des parties complètes d'activités, sans découper les activités même en multiples étapes opérationnelles.

Organiser la navigation à l'intérieur des formulaires suivant un ordre de tabulation logique et cohérent. Le formulaire doit avoir du sens aussi bien quand il est lu de façon linéaire par le lecteur d'écran que quand il est consulté à travers la touche de tabulation. Si nécessaire, utiliser l'attribut TABINDEX du tag <INPUT> pour donner un ordre de tabulation correct.

Organiser le flux de l'interaction à l'intérieur du formulaire de façon verticale (du haut vers le bas) ou de façon horizontale (de gauche à droite ou de droite à gauche, en fonction du sens de lecteur de la langue utilisée) sur la base des contenus qui doivent être représentés dans la page et aussi de la taille et du nombre de champs/contrôles.

Éviter les listes très longues de boutons radio ou de cases à cocher (maximum 5 boutons radio ou 5 cases à cocher), et utiliser éventuellement un menu déroulant ou une liste à sélection multiple.

Aligner les boutons radio et les cases à cocher de préférence verticalement. Rappelons que, dans un ensemble de boutons radio, l'un d'eux doit être sélectionné par défaut.

Toujours pour garantir l'indépendance du dispositif d'input, tous les éléments du formulaire doivent être utilisables également à partir du clavier. Les touches à utiliser sont le tab, le shift-tab, les flèches, la barre d'espace et l'envoi. Pour cette raison, comme nous l'avons indiqué au paragraphe 2.4.2, il ne faut pas permettre la génération automatique d'événements au moment où l'on quitte le focus d'un champ ou d'un contrôle (par exemple à travers l'instruction Onchange et OnSelect pour la souris) : il faut toujours insérer un bouton de confirmation explicite de l'action.

Si nécessaire, mentionner les champs obligatoires avec un astérisque placé à la fin de l'étiquette (exemple : Prénom* :) en spécifiant au bas du formulaire le sens de l'astérisque.

Il est préférable de ne pas utiliser les tableaux pour disposer les champs sur le formulaire, mais plutôt de recourir aux feuilles de style (CSS).

Ne pas utiliser les listes déroulantes pour l'insertion de la date : utiliser les champs d'input préformatés de façon appropriée.

Référence WCAG 2.0 : 1.4.2 ; 1.4.8 ; 2.1.1 ; 2.1.2 ; 2.4.3 ; 3.2.5

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.a et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 34 ; 36

b. Étiquettes (label)

Tous les champs doivent avoir une étiquette (label).

Les étiquettes et les champs doivent être alignés à gauche ou à droite selon le sens de lecture de la langue (côté gauche pour les langues occidentales, côté droit pour la langue arabe). Le champ doit être placé le plus près possible des étiquettes afférentes, en cherchant à respecter également l'alignement entre les champs eux-mêmes.

Les étiquettes :

- Doivent être associées sans équivoque aux contrôles relatifs en utilisant nécessairement le tag <LABEL FOR> et l'attribut ID du tag <INPUT> ;
- Doivent être claires, concises, significatives et descriptives ;
- Doivent être placées toujours sur la même ligne des contrôles auxquels elles se réfèrent, de préférence devant les champs et non au-dessus, à l'exception des boutons radio et des cases à cocher, qui prévoient l'étiquette toujours sur la même ligne, mais après le champ ;
- Ne doivent pas être trop éloignées des champs relatifs, pour favoriser les malvoyants qui utilisent des logiciels d'agrandissement qui sont donc liés à une visualisation très limitée de l'écran ;
- Doivent être cohérentes, c'est-à-dire qu'il faut utiliser toujours la même étiquette pour identifier le même champ ;
- Doivent se terminer par un double point (par exemple, « Prénom du contribuable : ») ;
- utiliser les caractères majuscules seulement pour la première lettre du premier mot de l'étiquette, sauf s'il s'agit d'un nom propre ou d'un acronyme.

Exemple de la manière de coder les étiquettes avec <LABEL FOR> et <INPUT ID> :

Dans ce formulaire, les étiquettes sont associées aux champs avec <LABEL FOR> et avec ID de <INPUT>.
TABINDEX est également établi pour assurer la navigation correcte entre les champs.

Nom: Prénom:

Grade: Marque:

Entreprise: Rôle:

Le code de cette étiquette est le suivant :

```
<form name="CV" … >
<label for="Nom">Nom:</label>
<input type="text" value="" id="Nom">
<label for="Pren"> Prénom:</label>
<input type="text" value="" id="Pren">
<label for="TitStudio">Grade:</label>
<select id="TitStudio">
 .....

</form>
```

Référence WCAG 2.0 : 3.1.5 ; 3.2.4 ; 3.3.2 ; 4.1.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes 3.2.2 et 3.2.4 selon la méthodologie indiquée dans le paragraphe 3.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 32 ; 35

c. FIELDSET et LEGEND, OPTGROUP

Il peut s'avérer utile de regrouper les contrôles à l'intérieur de cellules semblables au group box des interfaces client/serveur. Avec (x)HTML, le même effet est obtenu avec les tag <FIELDSET> et <LEGEND>.

En effectuant ce choix, il est nécessaire de tenir compte du fait que, dans certaines conditions, les lecteurs d'écran lisent le texte à l'intérieur de <LEGEND> avec l'étiquette du champ. Dans l'exemple qui suit, le lecteur d'écran lit : « données personnelles nom édition données personnelles prénom édition formation titre d'étude case à cocher formation mention édition... ».

Le <FIELDSET> doit donc être utilisé quand il est effectivement utile pour regrouper des champs et des contrôles qui font partie d'une même catégorie et qui, lus séparément, peuvent apparaître ambigus.

Il ne faut pas utiliser le <FIELDSET> pour créer des effets de formatage de la page car, outre qu'il

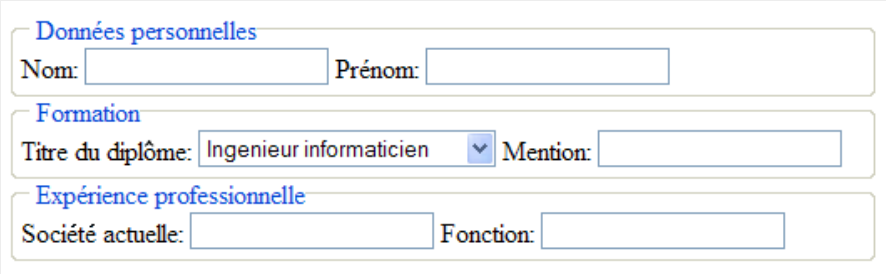
s'agit d'un usage impropre de ce contrôle, comme on l'a déjà dit, le lecteur d'écran lirait le LEGEND avant tout autre contrôle dans le FIELDSET.

En cas de listes très longues et/ou relevant de typologies différentes, il peut être utile de recourir au tag <OPTGROUP> pour séparer les mots d'une liste déroulante ou d'une boîte combinée en plusieurs groupes en attribuant un titre à chaque groupe.

Exemple de la manière d'organiser correctement le <FIELDSET> :

Dans ce formulaire, les champs sont regroupés avec <FIELDSET> et <LEGEND>. Les étiquettes sont associées au champ de manière univoque au moyen de <LABEL FOR> et ID.

TABINDEX assure la tabulation correcte. De plus, <OPTGROUP> est présent dans <SELECT>.



Le code de ce formulaire est le suivant :

```
<form name="CV1">
  <fieldset>
    <legend>Données personnels</legend>
      <label for="Nom">Nom:</label>
      <input type="text" tabindex="1" value="" id="Nom">
      <label for="Prenom">Prénom:</label>
      <input type="text" tabindex="2" value="" id="Prenom">
  </fieldset>
  <fieldset>
      <legend>Formation</legend>
        <label for="TitDiplome">Titre du diplome:</label>
        <select id="TitDiplome" tabindex="3">
          <optgroup label="Titre">
            <option value="L1">Ingenieur informaticien</option>
            <option value="L2">Ingenieur electronic</option>
          </optgroup>
          <optgroup label="Diplome">
            <option value="D1">Maitrise</option>
            <option value="D2">Mastere</option>
          </optgroup>
          <optgroup label="Autre">
            <option value="A1">Licence</option>
```

```
            <option value="A2">Doctorat</option>
          </optgroup>
        </select>
        <label for="Mention">Mention:</label>
        <input type="text" tabindex="4" value="" id="Mention">
  </fieldset>
  …..

</form>
```

Référence WCAG 2.0 : 3.3.2 ; 4.1.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 33

d. Boutons

Comme indiqué au paragraphe II.4.b, la génération automatique d'événement lorsque la souris quitte un champ (par exemple avec les fonctions Onchange et OnSelect de la souris) n'est pas consentie afin d'assurer l'accessibilité. Tous les formulaires doivent donc avoir au moins un bouton.

Placer les boutons d'action au bas du formulaire ou près des contrôles pour lesquels une confirmation de l'utilisateur est requise. Les boutons doivent être des boutons standard de Windows (OK).

Il est fortement déconseillé d'utiliser des boutons (graphiques) codés en tant que liens, car les utilisateurs de lecteurs d'écran ne parviennent pas à les distinguer des liens de navigation normaux et à les activer sans les avoir atteints avec le tabulateur.

Lorsqu'il est nécessaire d'avoir un bouton à l'aspect graphique différent du bouton standard, utiliser les CSS pour en modifier l'apparence esthétique.

Les règles pour l'intitulé des boutons sont les suivantes :

- OK : accepte les changements effectués et active la transaction (à utiliser, par exemple, dans le formulaire d'identification de l'utilisateur) ;

- Rétablir : rétablit les valeurs et paramètres par défaut ;
- Nettoyer : supprime les données insérées par l'utilisateur ;
- Imprimer : pour imprimer une page.

Les boutons peuvent être adaptés au contexte s'il existe un risque d'ambiguïté ou pour mieux préciser l'action du système. Par exemple, dans les pages de paramétrages de la recherche, il est préférable d'utiliser « Rechercher », tandis que l'on peut utiliser « Envoyer » dans les pages d'envoi des messages.

Le bouton « Annuler » ne doit jamais être utilisé.

Les tailles des boutons doivent permettre de rendre clairement lisible l'étiquette qu'elles contiennent, par exemple en utilisant de manière appropriée la marge entre l'étiquette et les bords du bouton.

Il faut toujours définir un bouton par défaut qui doit être activable au moyen de la touche « Enter » du clavier.

Exemple de la manière de rendre accessible un bouton graphique :

```
<label for="recherche" class="skiplink">Cherche : </label>
<input type="text" id="recherche" name="ric" value="" >
<input type="image" class="image" alt="OK" src="img_arrow_white.gif" />
```

Cherche :

Référence WCAG 2.0 : 3.2.5 ; 3.3.2 ; 4.1.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 34 ; 37 ; 38 ; 39

e. Messages

Comme nous l'avons déjà indiqué au paragraphe II.4.a, si des scripts de validation côté client sont prévus, ces contrôles doivent être dupliqués côté serveurs dans la mesure où certains navigateurs ne supportent pas JavaScript.

Les messages d'erreur doivent être fournis de manière claire et précise, en indiquant le problème et les actions nécessaires pour réparer l'erreur (par exemple, « Le code fiscal n'a pas été introduit », plutôt que « Il manque une donnée obligatoire » ou « Le champ Code fiscal n'a pas été valorisé »).

Il faut maintenir les données déjà introduites en cas de signalement d'erreur et/ou de retour à la page du formulaire.

Référence WCAG 2.0 : 2.2.5 ; 3.3.3 ; 3.3.4 ; 3.3.6

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 40 ; 42

f. Durée des sessions

Si, pour bénéficier d'un service fourni sur une page, un laps de temps déterminé est prévu pour effectuer certaines actions. Il est nécessaire d'avertir les utilisateurs, car certains pourraient avoir besoin de plus de temps pour remplir un formulaire ou pour lire une information.

Dans ces cas, il y a lieu d'indiquer également le temps maximum utile, en fournissant d'éventuelles alternatives pour bénéficier du même service.

Référence WCAG 2.0 : 2.2.1 ; 2.2.3 ; 2.2.5

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 41

g. Captcha

Dans le domaine de l'informatique, l'acronyme CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) désigne un test fait d'une ou de plusieurs questions et réponses pour déterminer si l'utilisateur est un humain et non un ordinateur ou, plus précisément, un robot.

Les CAPTCHA se basent sur des images particulièrement déformées de lettres et/ou de numéros que l'utilisateur doit reconnaître et écrire sans erreur dans un champ approprié, pour achever avec succès une interaction, par exemple l'inscription à un service en ligne.

Dans la mesure où ces tests sont projetés spécifiquement pour ne pas être lisibles par des instruments automatiques, les outils d'assistance technologiques normaux utilisés par les utilisateurs aveugles ou malvoyants ne sont pas capables de les interpréter. Dans certains cas, les images sont à ce point distordues qu'elles sont quasi impossibles à lire même pour des personnes jouissant d'une capacité de vision parfaite.

Afin de favoriser l'accessibilité, le W3C a mis à disposition une étude (http://www.w3.org/TR/turingtest/) qui illustre une approche différente de la réalisation des CAPTCHA : on n'utilise pas d'images graphiques déformées, mais des tests logiques basés sur des jeux de mots, des quiz mathématiques ou des demandes simples, qui peuvent empêcher l'accès aux robots.

Référence WCAG 2.0 : 1.1.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.a et III.2.b selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : chapitre «Observations additionnelles»

9. Tableaux de données

a. Caractéristiques générales

Les tableaux de données servent à visualiser des informations de type tabellaire où des relations horizontales et verticales claires et spécifiques sont établies entre les contenus des cellules.

Il ne faut pas utiliser les tableaux pour présenter des contenus dont la nature ne prévoit pas de comparaison entre lignes et colonnes, c'est-à-dire pour présenter en colonne des textes à lire sur une ligne, étant donné que les lecteurs d'écran interprètent toujours la première ligne comme le titre des colonnes situées en dessous.

Chaque tableau contient :

- des cellules dédiées aux en-têtes (tag <TH> Table Header), qui permettent de qualifier et d'identifier les données ;
- des cellules contenant les données (tag <TD> Table Date).
- Les tableaux peuvent être :
- simples, s'ils ont un seul niveau d'en-tête ;
- complexes, s'ils ont plusieurs niveaux d'en-tête (en-têtes multiples).

b. Titre des tableaux et des colonnes

Pour insérer le titre d'un tableau, dans (x)HTML, le tag à utiliser est <CAPTION>, car il positionne le titre en dehors du tableau lui-même.

La cellule d'en-tête d'une colonne, qui fournit le titre de la colonne elle-même, doit être codifiée avec le tag <TH> et non avec le tag <TD>, qui identifie une cellule de données. De cette manière, quand les lecteurs d'écran liront la donnée contenue dans une cellule, ils liront aussi la cellule

d'en-tête afférente, à laquelle la donnée se réfère. Dans l'exemple ci-après, quand l'utilisateur se trouvera, par exemple, sur la troisième ligne de la troisième colonne, le lecteur d'écran dit: « Flen Ben Foulen téléphone 71 710 170 ».

Le style du texte dans <CAPTION> et/ou dans <TH> peut être modifié en utilisant le CSS.

Il peut être utile de fournir une description des contenus du tableau à travers l'attribut SUMMARY du tag <TABLE>, au bénéfice des utilisateurs qui emploient des lecteurs d'écran.

Il faut toutefois se rappeler que le SUMMARY peut alourdir beaucoup la lecture de la page et donc la description à mettre dans le SUMMARY doit être très synthétique et fonctionnelle, comme indiqué dans les exemples précédents.

Si le tableau présente déjà un titre <CAPTION> correctement descriptif, le SUMMARY ne doit pas être mis.

Exemple de la manière d'organiser les titres dans le tableau des données :

Ceci est le titre du tableau inséré en utilisant le tag <CAPTION>		
Nom du contribuable	**Adresse**	**Téléphone**
Ali Ben Ali	5, rue du Jardin - Tunis	71 710 710
Flen Ben Foulen	2, Av Abou Al Kacem El Chabi -Tunis	71 710 170

Le code de ce tableau est le suivant:

```
<table summary="Le tableau reprend l'adresse et le numéro de téléphone des différents contribuables">
<caption> Ceci est le titre du tableau inséré en utilsant le tag &lt;CAPTION&gt;</caption>
<tr>
<th>Nom du contribuable</th>
<th>Adresse</th>
<th>Téléphone</th>
</tr>
<tr>
<td> Ali Ben Ali </td>
<td>5, rue du Jardin - Tunis </td>
<td>71 710 710</td>
</tr>
.....
</table>
```

Référence WCAG 2.0 : 1.3.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 43

Les cellules de la première colonne du tableau doivent contenir une valeur descriptive valable pour toute la ligne à partir du moment où les lecteurs d'écran considèrent les informations figurant dans la première colonne comme en-têtes des lignes respectives.

Deux exemples sont présentés ci-après, l'un est négatif et l'autre est positif, au sujet de la signification des données de la première colonne.

Exemple de tableau avec des données peu descriptives et des données suffisamment descriptives pour la première colonne :

Dans le tableau ci-dessous, les données de la première colonne sont peu descriptives. La cellule relative à l'adresse de Sfax, par exemple, est lue de la manière suivante par le lecteur d'écran : « colonne 3, ligne 3, adresse 2016, 64 Avenue Franklin Roosevelt ». Mettre le code postal dans la première colonne n'aide pas à comprendre de quelle ville il s'agit.

Code postal	Ville	Adresse	Téléphone
1080	Tunis	5, rue du Jardin - Tunis	71 353 444
2016	Sfax	64, Avenue Franklin Roosevelt	73 522 123
8050	Hammamet	10, Avenue Moncef Bey	72 222 777

Dans le tableau suivant, en revanche, quand le lecteur d'écran doit lire la même cellule que dans le tableau précédent, il lit comme suit : « colonne 3, ligne 3, adresse Sfax, 64, Avenue Franklin Roosevelt », en rendant compréhensible ce à quoi renvoie la donnée de la cellule sélectionnée.

Ville	Code Postale	Adresse	Téléphone
Tunis	1080	5, rue de Jardin - Tunis	71 353 444
Sfax	2016	64, Avenue Franklin Roosevelt	73 522 123
Hammamet	8050	10, Avenue Moncef Bey	72 222 777

Référence WCAG 2.0 : 1.3.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.b selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 44

c. En-têtes multiples

Lorsqu'il est nécessaire d'utiliser des tableaux avec des en-têtes multiples, où sont présentes différentes en-têtes en relation avec différents groupes de colonnes, la simple construction du tableau avec des cellules d'en-têtes <TH> et des cellules de données <TD> n'est pas suffisante en vue de permettre au lecteur d'écran de lire toutes les informations utiles pour comprendre la donnée. Dans ces cas, utiliser les attributs ID et HEADERS :

- ID permet d'identifier sans équivoque une cellule : il doit être utilisé pour identifier les cellules qui contiennent les données descriptives des « cellules de données » (classiquement, les cellules des en-têtes et celles de la première colonne) ;
- HEADERS permet d'associer les différentes cellules entre elles en se basant sur l'ID des cellules.

Pour des exigences différentes, on peut utiliser également :

- AXIS : permet une spécification ultérieure de la donnée identifiée avec ID, par exemple la typologie de donnée ;

- ABBR : permet de faire dire au lecteur d'écran la forme étendue de l'abréviation ;
- ROWGROUP et COLGROUP : pour associer l'en-tête à des groupes de lignes ou à des groupes de colonnes ;
- THEAD, TBODY, TFOOT : pour diviser le tableau en sections logiques.

Exemple de tableau avec en-têtes multiples :

Résumé versements (titre inséré avec CAPTION)				
Données du contribuable		**Versements en dinars**		
Nom	**Code fiscal**	**1991**	**1992**	**1993**
Flan Ben Foulen	RRRRRR66D54I348B	220	305	250
Ali Ben Ali	BBBBBB00G54H501T	380	107	557

Dans ce cas, quand l'utilisateur se trouvera sur la ligne 3 de la colonne 3, le lecteur d'écran lira : « colonne 3, ligne 3, versements en dinars, années 1991, Flan Ben Foulen, 220 ».

Le code de ce tableau est le suivant :

```
<table summary="Le tableau contient un résumé des versements par les contribuables pour les années 1991, 1992, 1993">
<caption>Résumé versements</caption>
<tr>
<th id="label_int1" colspan="2">Données sur le contribuable</th>
<th id="label_int2" colspan="3">Versements en dinars</th>
</tr>
<tr>
<th id="label_nom">Nom</th>
<th id="label_cod">Code fiscal</th>
<th id="label_91" axis="année">1991</th>
<th id="label_92" axis="année">1992</th>
<th id="label_93" axis="année">1993</th>
</tr>
<tr>
<td id="label_rossi"> Flan Ben Foulen </td>
<td headers="label_int1 label_Flan label_cod">RRRRRR66D54I348B</td>
<td headers="label_int2 label_Flan label_91">220</td>
<td headers="label_int2 label_Flan label_92">305</td>
<td headers="label_int2 label_Flan label_93">250</td>
</TR>
…
</table>
```

Référence WCAG 2.0 : 1.3.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 45

10. Graphisme

a. Caractéristiques générales

Éviter de présenter des textes sous forme d'image parce que les images ne peuvent pas être agrandies de façon adéquate : les malvoyants ne les voient pas de façon adéquate et ils sont contraints de les désactiver. De plus, un texte en format image requiert nécessairement de le spécifier au moyen de l'attribut ALT du tag <IMG>.

S'il n'est pas possible de se passer des textes au format image, les images doivent être réalisées avec les critères de différenciation indiqués au paragraphe II.10.d.

Comme cela a déjà été dit au paragraphe II.8.d, il est fortement déconseillé d'utiliser des boutons graphiques, car les utilisateurs de lecteurs d'écran ne parviennent pas à les distinguer des liens de navigation normaux et à les activer sans les avoir atteints avec le tabulateur. Lorsqu'il est nécessaire d'avoir un bouton à l'aspect graphique différent du bouton standard, utiliser les CSS pour en modifier l'apparence esthétique.

Comme indiqué au paragraphe II.5.b, il faut prêter une attention particulière à l'impact des images (photos, titres de pages, etc.) sur la possibilité de régler les dimensions de la page et de l'adapter à l'interface utilisée par les utilisateurs.

Ne pas présenter des objets clignotants ou en mouvement dont les fréquences d'intermittence peuvent provoquer des troubles d'épilepsie photosensible, des troubles de la concentration ou qui peuvent causer le mauvais fonctionnement des outils d'assistance technologique (Cf. Paragraphe II.6.b)

b. ALT

Il convient de fournir toujours un équivalent textuel pour chaque élément non textuel à travers l'attribut ALT du tag <IMG>.

Le texte alternatif spécifié dans ALT doit avoir une signification équivalente à celui transmis à travers les images, c'est-à-dire associée à la fonction exercée par l'objet original dans le contexte spécifique.

Le texte explicité à travers ALT ne devrait pas dépasser les 60 caractères.

Si l'image n'est pas significative et ne véhicule donc aucun contenu d'information (puces graphiques, images de remplissage, etc.), elle doit être gérée comme fond pour le CSS et non insérée dans la page (x)HTML. S'il y a lieu de l'ajouter dans la page, l'ALT doit être codé de cette manière : ALT="".

L'équivalent textuel doit être fourni également pour les applets, scripts et fichiers audio et vidéo, à travers lesquels les personnes handicapées ne parviennent pas à interagir.

Éviter d'utiliser des images pour la création de lignes ou de barres de séparation et réaliser ces barres au moyen du tag correct <HR>. Pour les accompagner de données graphiques, utiliser les CSS.

Exemple de ALT sur des images d'importance diverse :

- pour des images de faible importance :
- pour des images d'importance moyenne :
- pour des images significatives (link) :

Référence WCAG 2.0 : 1.1.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 46 ; 47 ; 51

c. LONGDESC

Pour les contenus complexes tels que graphiques, applets, là où le texte avec l'attribut ALT peut apparaître insuffisante, on peut fournir une description additionnelle en utilisant l'attribut LONGDESC, qui propose un lien à un fichier HTML où une description plus détaillée de l'objet peut

être insérée.

Cependant, dans la mesure où LONGDESC n'est pas encore supporté à l'heure actuelle par tous les navigateurs, il doit être accompagné d'un lien explicite vers le fichier HTML qui comprend la description de l'objet.

En tous les cas, compte tenu de la disponibilité d'espace sur la page, il est conseillé de mettre les contenus du fichier descriptif directement sur la page, sans renvoyer à ceux-ci à travers un lien.

Exemple de la manière de régler LONGDESC :

Référence WCAG 2.0 : 1.1.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 46

d. Couleurs et contrastes

Tous les éléments d'informations et toutes les fonctionnalités transmises à travers la couleur doivent être disponibles même en l'absence de la couleur particulière utilisée pour les présenter. La couleur ne saurait donc être utilisée comme unique forme de communication. Par exemple :

- Pour mettre en évidence un mot, ne pas utiliser uniquement le changement de couleur, mais utiliser aussi le gras ou une autre forme graphique ;
- Pour mettre en évidence les champs obligatoires, ne pas utiliser la couleur (par exemple « les champs obligatoires sont marqués en rouge ») mais ajouter plutôt un astérisque à la fin de l'étiquette ;
- Eviter les phrases du genre « les données marquées en rouge sont erronées », mais fournir un message adéquat qui explique avec clarté la nature de l'erreur.

Ne pas utiliser de couleurs fluorescentes et ne pas présenter des couleurs voisines appartenant aux extrêmes opposés de la gamme de couleurs (par exemple, bleu et noir).

Utiliser toujours un contraste fort en termes de luminosité et de couleur entre le fond et le texte, en se basant sur l'algorithme fourni par la W3C, de manière à être facile à distinguer par des personnes ayant différents problèmes de perception de la couleur. Pour ces motifs, il est toujours préférable d'utiliser le texte sur fond blanc ou de tonalité très claire. Éviter d'utiliser des fonds lignés, marbrés, en dégradé, etc.

Référence WCAG 2.0 : 1.4.1 ; 1.4.3 ; 1.4.6

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d'accessibilité : 18 ; 19

e. Plans

Si un plan sensible est utilisé pour la navigation ou la présentation des liens, celui-ci doit être défini côté client et non côté serveur. Il convient de fournir ce type de plans :

- un attribut ALT sur l'image du plan ;
- un attribut ALT également sur les zones sensibles (liens) du plan.

L'ALT doit être significatif et mesure facilement la compréhension des contenus des pages de manière à ce que l'utilisateur puisse clairement comprendre ce qu'il trouvera dans les différentes pages avant même d'y accéder. Par exemple, si le plan est celui de la ville de Tunis et les zones sensibles sont les quartiers :

- un ALT doit être placé sur l'image de Tunis comme, par exemple, « Liste des arrondissements de Tunis », et non « cliquez sur un arrondissement», qui est plus vague ;
- toutes les zones sensibles doivent avoir un ALT comportant le nom des différents

arrondissements.

Il faut tenir compte du fait que, lorsque les images sont désactivées par les utilisateurs, comme le font souvent les malvoyants en raison de leur difficulté à lire les images dont ils ne peuvent régler la taille), le plan deviendrait totalement inaccessible. Donc, sur la même page que le plan, il faut fournir la liste textuelle des liens accessibles à travers les zones sensibles : concrètement, les liens doivent être dédoublés.

Si, pour quelque motif que ce soit, il est indispensable d'utiliser un plan côté serveur, il est cependant nécessaire de présenter la liste textuelle des liens activables à travers le plan.

Exemple de carte avec liens dupliqués sur la même page :

```
<img src="Librairie.gif" usemap="#map1" alt="Secteurs de la librairie" />

<map id="map1" name="map1">

 <area shape="rect" coords="0,0,30,30" href="essais.html" alt=" Essais“ />

 <area shape="rect" coords="34,34,100,100" href="romans.html" alt="Romans" />

 <area shape="rect" coords="55,55,200,20000" href="manuels.html" alt="Manuels" />

</map>
<ul>

<li><a href=" essais.html">Essais</a></li>

<li><a href=" romans.html">Romans</a></li>

<li><a href=" manuels.html">Manuels</a></li>

</ul>
```

Référence WCAG 2.0 : 1.1.1

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 48 ; 49

11. Documents non html

a. PDF

L'utilisation de documents PDF est très répandue sur le web, pour divers besoins. Ceci a toutefois créé d'importantes barrières pour l'accessibilité, étant donné que les PDF réalisés de manière incorrecte ne s'intègrent pas bien avec les outils d'assistance technologique.

Nous nous référons en particulier aux PDF qui proviennent de la numérisation d'un document papier. Ce type de contenus (image provenant de la numérisation) doit être absolument évité : les outils d'assistance technologique n'arrivent en effet pas à les lire.

Si l'on doit partir obligatoirement d'un original sur support papier, avant de le convertir en PDF, il faut utiliser un programme de reconnaissance des caractères (OCR).

Il est par contre possible de rendre facilement accessible un PDF dérivant d'un document électronique (Word ou Open Office) en utilisant Acrobat Professional d'Adobe, selon les indications fournies par le producteur, comme indiqué à la page http://www.adobe.com/accessibility/products/acrobat/training.html.

Il faut indiquer clairement dans le texte du lien le type de document que l'on est sur le point d'ouvrir (pdf).

Il peut également être utile de fournir le lien pour télécharger le plugin, pour les utilisateurs qui ne le possèdent pas.

Exemple de lien vers document PDF :

 Résultats du contrôle automatisé des déclarations – format pdf

Référence WCAG 2.0 : 4.1.2

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.b selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : moyen

Objet de la check-list d'accessibilité : 50

b. Flash

Adobe Flash est un logiciel qui permet de créer des animations vectorielles principalement pour le web. Il est utilisé en outre pour créer des sites/applications entiers et, grâce à l'évolution des dernières versions, il est devenu un instrument puissant pour la création de Rich Internet Application et de plateformes de streaming audio/video.

Pour Flash également, comme pour les PDF, d'importantes barrières se sont créées pour l'accessibilité, dans la mesure où les fichiers .swf, extension qui caractérise les fichiers réalisés en Flash, réalisés de manière incorrecte ne s'intègrent pas bien avec les outils d'assistance technologique.

La pratique la plus correcte pour l'accessibilité est donc de ne pas les utiliser. Si l'on ne peut pas les éviter, il faut les rendre directement accessibles en suivant les lignes directrices sur l'accessibilité du producteur, comme indiqué à la page http://www.adobe.com/accessibility/products/flash/ ou fournir, pour les vidéos, une alternative textuelle tel qu'il est décrit au paragraphe I.11.3.

Comme pour les PDF, il est nécessaire d'indiquer clairement dans le texte du lien le type de document que l'on ouvre (flash) et de fournir le lien pour télécharger le plugin, pour les utilisateurs qui ne le possèdent pas.

Exemple de lien vidéo réalisé avec Flash :

Colloque à l'hôtel Excelsior – vidéo au format flash

Compte rendu textuel du colloque à l'hôtel Excelsior

Référence WCAG 2.0 : 4.1.2

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans les paragraphes III.2.b et III.2.c selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : difficile

Objet de la check-list d'accessibilité : 46 ; 50 ; 52

c. Fichiers audio/vidéo

Rendre accessibles de façon complète les contenus d'un film ou d'une présentation multimédia peut être difficile à partir du moment où la communication audio/vidéo fait appel à des sens qui sont déficitaires chez les personnes handicapées de la vue et de l'ouïe.

La décision sur le type d'intervention afin de rendre ces contenus accessibles dépend :

- De la typologie du contenu d'information ;
- De l'importance du contenu d'information ;
- Des difficultés de réalisation (surtout dans le cas de présentations en temps réel).

On considère qu'un « média » (audio ou vidéo) est indispensable quand il n'est pas possible de comprendre la signification globale du contenu à transmettre si l'on n'a pas accès à ce média.

En relation avec ces aspects, et en considération des outils d'assistance technologique (lecteur d'écran), on peut imaginer les cas suivants avec les interventions correspondantes possibles pour implémenter l'accessibilité pour les aveugles et les sourds :

- Film où le son et l'image sont interdépendants (la vidéo n'est pas décrite par une voix) :
    - transcription de la voix sur la page html, avec description des éventuelles parties importantes de la vidéo,
    - réalisation de légendes synchronisées avec la vidéo ;
- Film où le son est explicatif de la partie vidéo (l'image est décrite par une voix) :
    - transcription de la voix sur la page html ou version audio uniquement ;
    - réalisation de sous-titres synchronisés ;
- Film où seule la partie audio est importante (ex. discours, message ou interview) :
    - transcription de la voix sur la page html ;
    - réalisation de sous-titres synchronisés.

Les standards pour les présentations synchronisées sont le SMIL de W3C et le SAMI de Microsoft basé sur Windows Media Player. Quicktime, Realplayer et Flash permettent d'ajouter des sous-titres aux vidéos.

Exemple de code d'un fichier de type .smi pour souligner une vidéo :

```
……..
<Sync Start=17000>
<p class=”ENCC” id=Source>Debra
<p class=”ENCC”>Cette brève vidéo vous montrera un exemple de l’utilisation des sous-titres. <SpanID:17000;>
<Sync Start=21500>
<p class=”ENCC” id=Source>Debra
<p class=”ENCC”>Cette méthode est fondamentale pour permettre aux personnes sourdes de profiter d’une vidéo.<SpanID:21500;>
……
```

Référence WCAG 2.0 : 1.2.1 ; 1.2.2 ; 1.2.3 ; 1.2.4 ; 1.2.5 ; 1.2.6 ; 1.2.7 ; 1.2.8 ; 1.2.9

Type de contrôle :

☐ Test automatique

☒ Évaluation humaine

Procédure de contrôle : utiliser les outils indiqués dans le paragraphe III.2.a selon la méthodologie indiquée dans le paragraphe III.3.

Difficulté du contrôle : facile

Objet de la check-list d’accessibilité : 51

III- vérification d'accessibilité basée sur la check-list

La mise en conformité d’un site/service existant trouve son fondement dans les activités de vérification de l’accessibilité et de l’utilisabilité, qui sont des phases essentielles de la conception centrée sur l’utilisateur et du processus pour la définition de la qualité globale du site/service.

L’évaluation basée sur des check-lists consiste à effectuer (par un expert d'accessibilité) une série de navigations sur le site/service à évaluer, en suivant une grille de référence. Cette méthodologie a une double valeur :

- Il s’agit d’une méthode efficace et économique pour certifier le niveau d’accessibilité et d’utilisabilité d’un site (nouveau ou existant) lorsque l’on ne dispose pas d’utilisateurs finaux à impliquer dans l’évaluation ;
- C’est une méthode qui remplace la vérification avec des utilisateurs, afin de certifier

préventivement que les utilisateurs peuvent effectivement accéder aux contenus et donc exécuter les activités.

Dans ce qui suit nous présentons la check-list, les outils que l'expert utilise pour répondre aux items de la check-list et la méthodologie d'application.

1. Check-list d'accessibilité

Pour faciliter et standardiser le processus d'évaluation d'accessibilité et d'utilisabilité d'un site web par l'expert, une check-list d'accessibilité a été mise au point, qui est disponible à l'annexe 1.

Une check-list se compose de 53 items, qui correspondent à 53 contrôles, à travers lesquelles l'expert en accessibilité exprime la conformité de l'interface avec les lignes directrices énoncées dans le chapitre 2 de ce document.

La check-list se divise en trois sections :

- introduction ;
- contrôles d'accessibilité ;
- observations supplémentaires.

La première section définit :

- le nom du site/de l'application web ;
- l'URL de référence ;
- les utilisateurs ciblés, c'est-à-dire savoir si le site/l'application s'adresse au personnel interne ou externe de l'administration de référence. Connaître les utilisateurs et le contexte dans lequel ils opèrent permet d'évaluer le contrôle sur les spécificités de l'environnement cible ;
- Outils utilisés pour l'évaluation (navigateur, outils d'assistance technologique, etc.).

La deuxième section reprend tous les items à vérifier par l'expert en accessibilité et qui subdivisés par typologie de contrôle,. Chaque item prévoit 3 possibilités :

- Oui, dans le cas où la caractéristique d'interface analysée est correspondante à tout ce qui a été prescrit par l'item ;
- Non, dans le cas où il ne correspond pas ; il faut alors mentionner dans le champ « Notes » les éventuelles indications qui peuvent aider le développeur à résoudre le problème ;

- N/C, dans le cas ou l'item n'est pas applicable parce que l'interface ne présente pas cette caractéristique (par exemple, aucune inscription en mouvement n'est présente sur le site).

Dans la troisième section, celle des « Observations supplémentaires », il est possible d'insérer toutes les indications additionnelles par rapport aux items de la section précédente, qui peuvent être utiles au développeur pour mieux comprendre les corrections et les améliorations qu'il devra réaliser.

L'évaluation sera positive seulement dans le cas où tous les items auront une réponse positive ou N/C.

2. Outils d'aide à l'évaluation

Il existe une variété d'outils que l'expert en accessibilité peut utiliser pour déterminer si un site web répond à des critères de conformité d'un site accessible, c'est-à-dire pour répondre aux items de la check-list.

En tout état de cause, aucun instrument d'évaluation isolé ne fournit des informations complètes ou est en mesure d'appréhender l'ensemble des problèmes relatifs à l'accessibilité d'un produit logiciel. Par exemple, le service en ligne du W3C pour la validation du code HTML permet de contrôler si le code est écrit correctement, mais il n'est pas en mesure de fournir une évaluation adéquate sur tous les détails et les éléments qui servent à la création d'un site accessible. Il ne fournit pas, par exemple, de réponse quant au caractère adéquat de l'attribut ALT des images. Pour répondre de façon adéquate, l'expert doit recourir à d'autres outils qui lui permettent de voir l'image dans le contexte où elle s'insère et, à partir de là, juger du caractère adéquat de l'ALT auquel elle est associée.

Le W3C/WAI fournit une liste précise, régulièrement mise à jour, de ces outils à la page « Web Accessibility Evaluation Tools » (http://www.w3.org/WAI/ER/tools/#Evaluation).

Nous illustrerons brièvement ci-après les principaux outils d'aide pour l'évaluateur.

a. Navigateur

Le navigateur joue un rôle central pour l'accessibilité web.

Les recommandations du chapitre II prévoient une conception cross-browser, c'est-à-dire non liée à un navigateur spécifique, pour faire face aux différentes exigences des utilisateurs.

Il devient donc important, au moment de la vérification, de contrôler la page avec différents navigateurs graphiques, dans les différentes versions et dans les différents systèmes

opérationnels et, dans le cadre de chaque typologie de navigateur, de contrôler s'il est possible de réaliser, avec toutes les personnalisations qui sont d'habitude mises en œuvre par les utilisateurs valides et handicapés :

- l'agrandissement des caractères par défaut ;
- la désactivation de JavaScript ;
- la navigation sans CSS ou le remplacement des CSS par ceux mis aux points par l'utilisateur ;
- la désactivation des images ;

La connaissance approfondie de la manière dont « travaille » un navigateur et dont il peut être personnalisé devient donc importante non seulement pour ceux qui doivent projeter et réaliser les pages, mais aussi pour l'expert en accessibilité.

Les navigateurs à prendre en considération sont indiqués au paragraphe II.5.c.

b. Outils d'assistance technologique

Les outils d'assistance technologique sont les instruments et les solutions techniques, matérielles et logicielles, qui permettent à une personne handicapée de surmonter ou de réduire les conditions de désavantage pour accéder aux informations et aux services fournis par les systèmes informatiques.

Il est donc nécessaire que l'expert en accessibilité ait une connaissance approfondie sur la manière d'utiliser ces technologies afin d'assurer que les sites/applications soient techniquement accessibles également à travers celles-ci.

- **Lynx**

  Lynx est un navigateur qui visualise uniquement les contenus textuels des pages. Il ne supporte pas la souris, les images (montrant donc le texte alternatif), les vidéos, audio, animations, plug-in, applet java et javascript.

  Il a été inséré dans les outils d'assistance technologiques parce qu'il est typiquement utilisé par des personnes ayant une perte partiels ou totale de la vue, avec le support d'un lecteur d'écran.

- **Lecteurs d'écran**

  Un lecteur d'écran est une application logicielle qui identifie et interprète le texte montré à l'écran d'un ordinateur, par synthèse vocale ou à travers un écran braille, et qui est

utilisée par des personnes ayant une perte partielle ou totale de la vue.

L'un des lecteurs d'écran les plus utilisés est Jaws (Job Access With Speech), produit par Freedom Scientific (www.freedomscientific.com).

Jaws est probablement le meilleur lecteur d'écran, non seulement pour la quantité de fonctionnalités qui l'enrichissent, mais parce qu'il permet d'utiliser la plus grande partie des applications Windows, comme Microsoft Office (Word, Excel, Access, PowerPoint, Outlook), Internet Explorer, etc.

Ce logiciel, payant, peut être installé seulement sur des systèmes d'exploitation Microsoft. Il existe une version démo, gratuite, qui permet d'utiliser Jaws pour 40 minutes maximum.

Il existe également des lecteurs d'écran limités à la navigation web, comme Home Page Reader, un produit d'IBM, mais moins répandu justement parce qu'il n'est utilisé que pour naviguer sur internet et lire le courrier électronique.

- **Agrandisseur d'écran**

  C'est un logiciel qui agrandit l'écran et visualise la portion d'écran où se trouve le curseur, comme s'il s'agissait d'une sorte de loupe. L'utilisateur se sert de la souris ou du clavier pour explorer l'écran à la recherche des éléments qui l'intéressent.

  Les agrandisseurs d'écran sont utilisés normalement par les malvoyants graves, c'est-à-dire ceux qui ont une réduction de la vue très élevée.

  Tous les agrandisseurs de nouvelle génération ont une vaste gamme d'outils de navigation, agrandissent les caractères et disposent d'une synthèse vocale pour aider la consultation de longues pages de texte.

  Parmi les agrandisseurs d'écran les plus utilisés, il y a ZoomText, produit par Ai Squared (www.aisquared.com) et Magic, de Freedom Scientific (www.freedomscientific.com).

c. Outils d'évaluation automatique du code

Ces outils permettent de vérifier si les pages web sont complètement adhérentes aux spécifications techniques du W3C pour le code HTML, XHTML et CSS. Ces outils sont :

- W3C's HTML Validation Service (http://validator.w3.org/) ;
- W3C's CSS Validation Service (http://jigsaw.w3.org/css-validator/) ;
- W3C Link Checker (http://validator.w3.org/checklink) : instrument de validation en ligne qui permet la vérification des liens et des ancres à l'intérieur des pages individuelles ou de

tout le site web ;

- Validatore del Web Design Group (http://www.htmlhelp.com/tools/validator/) : évalue la conformité des pages aux spécifications du W3C, comme le valideur du W3C, avec un maximum 100 pages à la fois ;

- Multipage Validator (http://www.validator.ca/) : évalue la conformité des pages aux spécifications du W3C, comme le valideur du W3C, avec un maximum 100 pages à la fois.

d. Outils d'évaluation semi-automatique

Parmi les outils d'évaluation semi-automatique particulièrement utiles, on trouve :

- **Web Accessibility Toolbar**

  Cet outil est fourni par la Web Accessibility Tools Consortium et peut être téléchargé à l'adresse web http://www.wat-c.org/tools/index.html.

  C'est un programme qui installe une série de boutons sous la barre de navigation d'Internet Explorer. Il a été créé dans le but de supporter la vérification d'un grand nombre d'aspects concernant l'accessibilité d'une page web, dont :

  - le code (x)HTML et CSS de la page courante, en utilisant les services en ligne de validation du code du W3C ;
  - la structure sémantique des tags (x)HTML ;
  - les tableaux simples et complexes, permettant également leurs linéarisation ;
  - l'activation/désactivation de CSS ;
  - l'activation/désactivation de JavaScript, plug-in et ActiveX ;
  - la modification de la taille de la fenêtre du navigateur ;
  - le contrôle du contraste texte/fond, au moyen de Color Contrast Analyser, décrit dans le paragraphe suivant ;
  - la visualisation du tag <IMG>, avec la description relative de l'attribut ALT.

- **Color Constrast Analyser**

  C'est un outil pour contrôler les combinaisons entre les couleurs de fond et le premier plan afin de vérifier si elles parviennent à garantir une bonne visibilité des couleurs. Il contient également les fonctionnalités pour vérifier si la combinaison satisfait ceux qui ont une perception altérée des couleurs, comme les daltoniens.

L'outil se base sur les algorithmes qui calculent la différence de couleur et la différence de luminosité suggérés par le W3C.

Il est installé en même temps que la Web Accessibilité Toolbar ou peut être utilisé indépendamment de celle-ci en le téléchargeant du site du Web Accessibility Tools Consortium (http://www.wat-c.org/tools/index.html).

3. Application de la check-list

Les étapes opérationnelles pour procéder à la vérification d'accessibilité au moyen de check-list sont :

- Connaître les thématiques sur l'accessibilité et l'utilisabilité, en référence particulière aux recommandations du chapitre 2 ;
- Identifier une sélection de page : la navigation à l'intérieur du site doit se faire en sélectionnant un échantillon représentatif des différents types de pages, par exemple :
  - la page d'accueil ;
  - Toutes les pages auxquelles l'utilisateur accède probablement en naviguant dans le site ;
  - Toutes les pages qui présentent des formulaires et les pages de réponse afférentes ;
- Effectuer une évaluation du code HTML, XHTML et CSS en utilisant les outils indiqués dans le paragraphe 3.2.3 (exécuter au moins un des outils de validation sur les 100 premières pages du site web) ;
- Sélectionner au moins trois différents navigateurs graphiques (par exemple Internet Explorer, Firefox, Opera), parmi ceux indiqués dans le paragraphe 2.5.3, en différentes versions, installées sur différentes plateformes (Windows, Linux, Mac) et vérifier que :
  - le contenu d'information et les fonctionnalités présents sur une page soient les mêmes dans les différents navigateurs ;
  - la présentation de la page soit semblable dans les différents navigateurs ;
  - le contenu d'information et les fonctionnalités de la page soient encore accessibles en cas de désactivation du chargement des images, en vérifiant si des textes alternatifs sont appropriés ;
  - les contenus d'information de fichiers audio éventuels soient accessibles, même sous forme textuelle ;

- les contenus de la page soient accessibles en cas d'utilisation des fonctions prévues par les navigateurs pour définir la grandeur des caractères ;
- les contenus des pages soient accessibles en cas de modification de la résolution de l'écran, en vérifiant que la barre de défilement horizontal ne soit pas activée ;
- la page soit consultable uniquement avec le clavier, en s'assurant de pouvoir accéder à tous les liens et à tous les contrôles des formulaires et que les liens indiquent clairement où ils mènent ;
- les contenus et les fonctionnalités de la page soient encore consultables, même selon des modalités différentes, en cas de désactivation des feuilles de style, script, applet et autres objets de programmation ;
- vérifier les différences de luminosité et de couleur entre le texte et le fond selon les algorithmes définis par le W3C ;
- modifier la couleur de l'écran en échelle de gris (ou imprimer la base en échelle de gris ou en noir et blanc) et observer si les liens sont identifiables visuellement ;

- Naviguer avec un navigateur textuel (Lynx) de manière à ce que les contenus et les fonctionnalités des pages sélectionnées maintiennent leur signification d'ensemble et leur structure sémantique correcte ;
- Naviguer avec un lecteur d'écran, par exemple Jaws, pour contrôler si :
  - il est possible de lire aisément la page ;
  - une hiérarchie claire des titres est possible ;
  - la page est opérationnelle en se déplaçant entre les liens, les formulaires et les tableaux ;
  - la page est opérationnelle avec JavaScript ou d'autres programmes non-HTML comme Flash ;
  - contrôler si les textes alternatifs des images sont correctement contextualisés ;
  - lire des documents ou des programmes non-HTML, comme les PDF ;
- Remplir chaque item de la check-list de l'annexe 1 en indiquant s'il est conforme (Oui), s'il n'est pas conforme (Non) ou s'il n'est pas applicable (N/C) après avoir effectué le contrôle afférent.

Comme cela a été dit, l'évaluation sera positive seulement dans le cas où tous les items auront une réponse positive ou N/C.

Cet examen permettra de définir les éventuelles interventions de correction à réaliser. Celles-ci peuvent être :

- Des interventions simples et d'amélioration, qui peuvent aller de la modification du code à une intervention sur le graphisme du site ;
- Des interventions complexes, comme la réélaboration graphique de certaines zones du site, jusqu'au renouvellement de la conception du lay-out de page.

ANNEXE 1

Check-list d'accessibilité

I- Introduction

Ce document reprend les résultats du contrôle d'accessibilité ou la correspondance du site/de l'application avec les recommandations reprises dans le chapitre 2 du Guide Technique.

Nom du site/de l'application : _____

Adresse internet : http:// _____

Contrôle effectué le : _____

Usage :

interne à l'administration (Intranet)
externe à l'administration (citoyens, professionnels, autres administrations, etc.)

Indiquer les instruments et les technologies utilisés pour effectuer le contrôle :

Nom	Version

II- Contrôles d'accessibilité

Code HTML ou XHTML et CSS		OUI	NON	N/C
1.	Dans le cas de sites/applications réalisés avec une DTD de type Strict de l'HTML ou de l'XHTML, les pages sont validées par le validateur du W3C.			

2.	Dans le cas de sites/applications réalisés avec une DTD de type Transitional de l'HTML ou de l'XHTML : • les instructions relatives au code reprises dans les recommandations ont été bien suivies ; • les pages sont validées par le validateur du W3C.			
3.	Dans le cas de sites/applications déjà existants avec des frames et réalisés avec une DTD de type Frameset de l'XHTML ou de l'HTML : • les instructions relatives au code reprises dans les recommandations ont été bien suivies ; • les pages sont validées par le validateur du W3C.			
4.	Les instructions relatives à l'internationalisation établies dans les recommandations, dont l'insertion correcte des attributs lang, dir et charset, ont été suivies			
5.	Le code CSS est validé par le validateur du W3C.			
Notes :				
Mise en page de la page		**OUI**	**NON**	**N/C**
6.	La page d'accueil et les pages internes sont divisées en zones fonctionnelles bien distinctes et ont un contenu clair (ex. menu de service, menu principal, contenus principaux et contenus secondaires).			
7.	Le contenu et la fonctionnalité présents dans les pages sont identiques lorsqu'ils sont visualisés dans les différents navigateurs graphiques indiqués dans les recommandations.			
8.	La présentation des pages est identique dans les différents navigateurs graphiques.			
9.	Les contenus et les fonctionnalités des pages sont actifs et utilisables même lorsque les applets, scripts et autres objets de programmation sont inactifs ou qu'ils ne peuvent être supportés. Si ce n'est pas possible, ces mêmes contenus et ces mêmes fonctionnalités sont fournis de manière équivalente.			
10.	Dans le cas où seules les CSS sont utilisées pour la mise en page, en les désactivant, les contenus sont présentés dans le bon ordre et la page est compréhensible et fonctionnelle.			

11.	Dans le cas où les tableaux sont utilisés (même en association avec les CSS) pour la mise en page : • les instructions relatives aux tableaux de mise en page reprises dans les recommandations ont été bien suivies ; • en excluant tant les CSS que les tableaux de mise en page, les contenus sont présentés dans le bon ordre, la page est compréhensible et fonctionnelle.			
12.	Dans les différents navigateurs graphiques, en variant la résolution, la mise en page : • est relative ; • ne comporte pas de superpositions de contenu ; • ne fait pas apparaître la barre de défilement horizontale en résolution 800x600.			
13.	La taille des fontes par défaut : • est exprimée de manière relative et correspond aux instructions des recommandations ; • en la variant à travers des commandes disponibles dans les différents navigateurs, les contenus des pages ne se superposent pas et sont encore utilisables, malgré un agrandissement de 200% des caractères			
14.	Vérification de la navigation avec le navigateur textuel (Lynx) : • Les contenus et fonctionnalités sont disponibles (de manière équivalente) tout comme avec les navigateurs graphiques ; • les contenus des pages gardent leur signification d'ensemble et leur structure sémantique.			
Notes :				
Présentation des contenus		**OUI**	**NON**	**N/C**
15.	Utilisation d'un titre significatif des contenus (<H1>), cohérent avec le contenu du TAG <TITLE>.			
16.	Les titres <H1> …<H6> sont utilisés pour structurer le texte dans la page et respectent un ordre hiérarchique.			
17.	Afin de définir le contenu (paragraphes, listes, citations, etc.), les tags (x)HTML ont été utilisés de manière sémantiquement correcte, selon les instructions établies dans les recommandations.			
18.	Toutes les informations et les fonctionnalités véhiculées au travers de la couleur sont disponibles même sans cette dernière et ce même pour la couleur utilisée pour les liens.			
19.	Les couleurs du texte et le fond sont établis de manière à garantir un contraste suffisant, contrôlé par l'algorithme proposé par le W3C et ce même pour les problèmes de daltonisme.			

20.	Les textes au format image sont réalisés de manière à garantir un contraste texte/fond suffisant, contrôlé par l'algorithme proposé par le W3C, et ce, même pour les problèmes de daltonisme.			
21.	En cas d'objets et de textes en mouvement, défilant ou clignotant dont les fréquences d'intermittence peuvent provoquer des troubles d'épilepsie photosensible et des troubles de la concentration : • l'utilisateur est averti préventivement ; • il y a des méthodes prévues pour les éviter ; • on a vérifié qu'ils ne provoquent pas des dysfonctionnements des outils de technologie d'assistance. (note : animations absentes ou inoffensives = N/C)			
22.	En cas d'abréviations, d'acronymes, de texte en langue étrangère, etc., les instructions relatives au code reprises dans les recommandations ont été bien suivies.			
23.	Les CSS sont utilisées pour définir le style des textes (alignement, couleur, fond, etc.) selon les instructions des recommandations.			
Notes :				

Navigation et liens		**OUI**	**NON**	**N/C**
24.	Les liens présents dans les pages sont sélectionnables et activables, dans le bon ordre, à travers les commandes du clavier, de la technologie d'émulation du clavier ou au moyen de systèmes de pointage autres que la souris.			
25.	Chaque lien indique clairement sa destination, même s'il est isolé du contexte ou du texte qui le précède ou le suit.			
26.	Il n'y a pas de liens portant le même nom et menant à des pages différentes.			
27.	Les liens sont reconnaissables à la vue, sans devoir passer dessus avec la souris et sans l'aide de la couleur.			
28.	Au moins pour les liens autres que les menus, l'état « déjà visité » est mis en évidence.			
29.	S'il y a, en début de page, des groupes de liens ou des contenus communs à toutes les pages, un lien (même caché) est prévu pour sauter directement au contenu principal.			
30.	Dans une liste de liens, ces derniers sont éloignés les uns des autres, tant verticalement qu'horizontalement.			

31.	En cas d'ouverture de nouvelles fenêtres et de pop-up dans le navigateur : • les instructions relatives au code reprises dans les recommandations ont été bien suivies ; • l'utilisateur est averti du changement de cible.			
Notes :				
Formulaires		**OUI**	**NON**	**N/C**
32.	Chaque champ du formulaire a sa propre étiquette, indiquant le contenu, codifiée avec <LABEL for>.			
33.	En cas d'utilisation de <FIELDSET> pour regrouper les champs, les instructions établies dans les recommandations ont été suivies.			
34.	Il est possible d'interagir avec tous les éléments du formulaire (éditer les champs, sélectionner des entrées dans les listes, activer des boutons, etc.) uniquement via le clavier.			
35.	Dans le formulaire, en mode de lecture linéaire : • les étiquettes sont disposées à proximité de leurs champs respectifs ; • les étiquettes et les champs sont correctement associés.			
36.	La tabulation du formulaire est correcte.			
37.	Utilisation des boutons standards et non des liens.			
38.	Les boutons d'un formulaire sont éloignés les uns des autres, tant verticalement qu'horizontalement.			
39.	Les dimensions des boutons sont adaptées à l'étiquette de manière à rendre leur contenu clairement lisible.			
40.	Les messages d'erreur sont affichés de manière claire et précise.			
41.	En cas d'expiration de la session de la page, l'utilisateur en est averti de manière claire et explicite par l'indication du temps maximum consenti et les éventuelles alternatives pour jouir du service.			
42.	En présence de contrôles côté client (Javascript) de la bonne saisie sur clavier des champs, les contrôles sont répétés côté serveur.			
Notes :				
Tableaux de données		**OUI**	**NON**	**N/C**
43.	Les tableaux de données ont un titre codifié par <CAPTION> et/ou par SUMMARY, selon les instructions des recommandations.			

44.	Dans les tableaux de données, le lecteur d'écran associe les contenus de chaque cellule de données de manière compréhensible à leurs entêtes respectives et en particulier : • la première cellule de chaque colonne est codifiée avec le tag de titre <TH> ; • la première colonne du tableau renferme les données significatives pour la compréhensibilité des autres cellules ; • il n'y a pas de colonnes ou de lignes vides d'espacement.			
45.	Pour les tableaux de données qui ont au moins deux niveaux logiques d'entêtes (tableaux complexes), en plus de ce qui est prévu dans les articles précédents, il faut aussi utiliser les bons attributs afin d'associer les cellules d'entête aux cellules de données, comme indiqué dans les recommandations.			
Notes :				
Images et plans		**OUI**	**NON**	**N/C**
46.	Tous les objets non textuels sont accompagnés d'une alternative textuelle (ALT ou LONGDESC selon la complexité de l'image) de manière à ce qu'en les désactivant, il n'y ait pas de pertes de contenu, de fonctionnalité et de possibilité d'utilisation.			
47.	Toutes les images non significatives ont la valeur de l'attribut ALT vide (ALT="") ou elles sont chargées comme image de fond au travers des CSS.			
48.	Dans les plans image (côté client ou côté serveur), des liens alternatifs visibles sont présents sur la page, de manière à ce que les fonctionnalités affichées sur le plan soient utilisables même lorsque les images sont désactivées.			
49.	Dans les plans image (côté client ou côté serveur), des textes alternatifs sont présents tant sur l'image du plan que sur les zones cliquables du plan.			
Notes :				
Documents non HTML		**OUI**	**NON**	**N/C**

50.	Les objets qui utilisent des technologies non définies par des grammaires formelles publiées (PDF, Flash, etc.) ont des scripts et des applets accessibles aux outils de technologie d'assistance.			
51.	En présence d'objets multimédia, des alternatives textuelles (x)HTML, un résumé bref, des légendes textuelles synchronisées et/ou des descriptions audio sont fournies selon l'importance du contenu.			
Notes :				

Script, applet et autres objets de programmation		**OUI**	**NON**	**N/C**
52.	Applet, script et autres objets de programmation, y compris les gestionnaires d'évènements, qui véhiculent de l'information ou qui requièrent des actions de la part de l'utilisateur sont indépendants du périphérique d'entrée utilisé (souris, clavier, etc.).			
Notes :				

Mise à jour automatique ou réacheminement de la page		**OUI**	**NON**	**N/C**
53.	En cas de mise à jour automatique ou de réacheminement (refresh ou redirect) de la page, défini dans la section <HEAD> de l'XHTML ou réalisés en Javascript : • les instructions relatives au code reprises dans les recommandations ont été bien suivies ; • en cas d'actualisation, l'utilisateur est averti de manière évidente et explicite.			
Notes :				

ANNEXE 2

Les handicaps à prendre en compte lors du développement des sites web

Il existe une grande littérature sur la manière dont les personnes saines interagissent avec les technologies informatiques. Celle-ci est rationalisée et désormais consolidée en un ensemble de principes qui représentent une référence dont il faut tenir compte dans la conception de l'interface et de l'interaction homme-machine.

À l'inverse, il y a moins de littérature concernant les modalités d'interaction des handicapés avec les technologies informatiques.

En fait, ces personnes n'interagissent pas souvent avec le produit informatique de manière directe. L'interaction est fournie par une interface : les outils d'assistance technologique utilisés pour compenser l'handicap qui les caractérise.

Par outils d'assistance technologique, on entend tout objet, faisant partie d'un équipement ou d'un système, produit de manière commerciale ou personnalisée, qui est utilisé pour augmenter, maintenir et améliorer les capacités fonctionnelles des personnes handicapées.

Dans ce contexte on prendra, uniquement, en considération les outils d'assistance technologique qui concernent l'interaction homme-machine et qui influence la manière avec laquelle l'utilisateur interagit avec l'ordinateur et, donc, avec le site/service spécifique mis en place.

La connaissance des handicapés, des outils d'assistance technologique qu'ils utilisent et de la manière dont ils les utilisent pour interagir avec l'ordinateur, devient fondamentale pour au moins deux aspects.

- D'un côté, il est nécessaire de connaître les outils d'assistance technologique afin de s'assurer que les produits/services réalisés leur soient techniquement accessibles.
- D'un autre côté, connaître les spécificités des utilisateurs (caractéristiques, exigences et préférences) dans leur interaction avec l'ordinateur permet de définir et d'implémenter cette flexibilité nécessaire qui garantit la jouissance des produits/services réalisés pour ces utilisateurs « spéciaux », aussi bien que pour tous les autres types d'utilisateurs.

I- Les personnes souffrant de troubles de la vision

Les troubles de la vision comprennent deux classes d'utilisateurs : les malvoyants (avec divers niveaux de gravité) et les aveugles.

Cependant, les problèmes d'accès à l'ordinateur sont différents dans les deux cas.

1. Les malvoyants

En général, les personnes malvoyantes perçoivent le texte et le graphisme de manière confuse et limitée. En réalité, comprendre pleinement une telle forme d'handicap n'est pas simple puisque différents facteurs interviennent :

- acuité visuelle (« combien je vois »)
- champ visuel (central, périphérique, maculaire, etc.)
- sensibilité aux couleurs
- sensibilité à la lumière
- qualité de la vision (brouillée, déformée, etc.).

L'interaction entre les différents facteurs mène à des états de perception extrêmement subjectifs et difficilement standardisables. Donc, selon la spécificité de leur handicap, pour interagir avec l'ordinateur, les malvoyants adoptent différentes méthodes et différents outils d'assistance technologique.

Dans quelques cas, ils utilisent des outils d'assistance technologique comme des agrandisseurs d'écran, souvent combinés à des lecteurs d'écran (synthèse vocale).

Dans d'autres cas, ils utilisent les fonctions de personnalisation du navigateur : ils augmentent la taille de la fonte, ils désactivent les feuilles de style (CSS - Cascading Style Sheets) du navigateur pour en charger d'autres avec des agencements personnalisés, ils changent les couleurs de la mise en page.

En résumé, les malvoyants ont besoin d'une importante flexibilité de l'interface afin qu'ils puissent l'adapter à leurs exigences, a priori, difficilement prévisibles.

Ces stratégies mettent en évidence l'insuffisante utilité des options d'agrandissement des caractères que certains développeurs mettent à disposition sur les pages web. Bien que l'attention soit appréciable et étant donné les nombreuses préférences dues aux différents défauts visuels, ces options ne satisfont presque jamais les attentes de l'utilisateur spécifique. Afin de favoriser ces types d'utilisateurs, ils implémentent d'autres caractéristiques comme indiqué dans le présent Guide Technique.

2. Les aveugles

Les aveugles ne peuvent pas percevoir les contenus visuels. Ils recourent donc aux outils d'assistance technologique qui exploitent les canaux sensoriels de l'ouïe et/ou du toucher, comme les lecteurs d'écran.

Les lecteurs d'écran sont des programmes logiciels qui identifient et interprètent le texte visualisé sur l'écran de l'ordinateur. Dans le cas où ces lecteurs utilisent une synthèse vocale, le contenu est lu à l'utilisateur (canal auditif). Dans le cas où un écran braille est utilisé, le contenu est rendu à l'utilisateur en symboles braille, se composant en relief sur une barre hardware que l'utilisateur touche avec les doigts (canal tactile).

Un utilisateur aveugle peut choisir d'utiliser un des deux systèmes ou les deux simultanément (synthèse vocale et écran braille) selon les circonstances.

Dans les deux cas, les contenus de l'écran sont présentés à l'utilisateur de manière « linéarisée », c'est-à-dire lus ou restitués en symboles Braille dans l'ordre dans lequel ils apparaissent dans le code de la page, indépendamment de la mise en page graphique.

Tous les lecteurs d'écran n'ont pas de caractères standards. Quelques-uns sont en mesure de lire des contenus présentés uniquement à travers des applications spécifiques. Évidemment, les lecteurs d'écran les plus répandus parmi les aveugles sont ceux qui sont les plus compatibles avec les applications les plus communément utilisées par les utilisateurs, et ce sont ceux qui assurent la garantie de la compatibilité des produits/services réalisés.

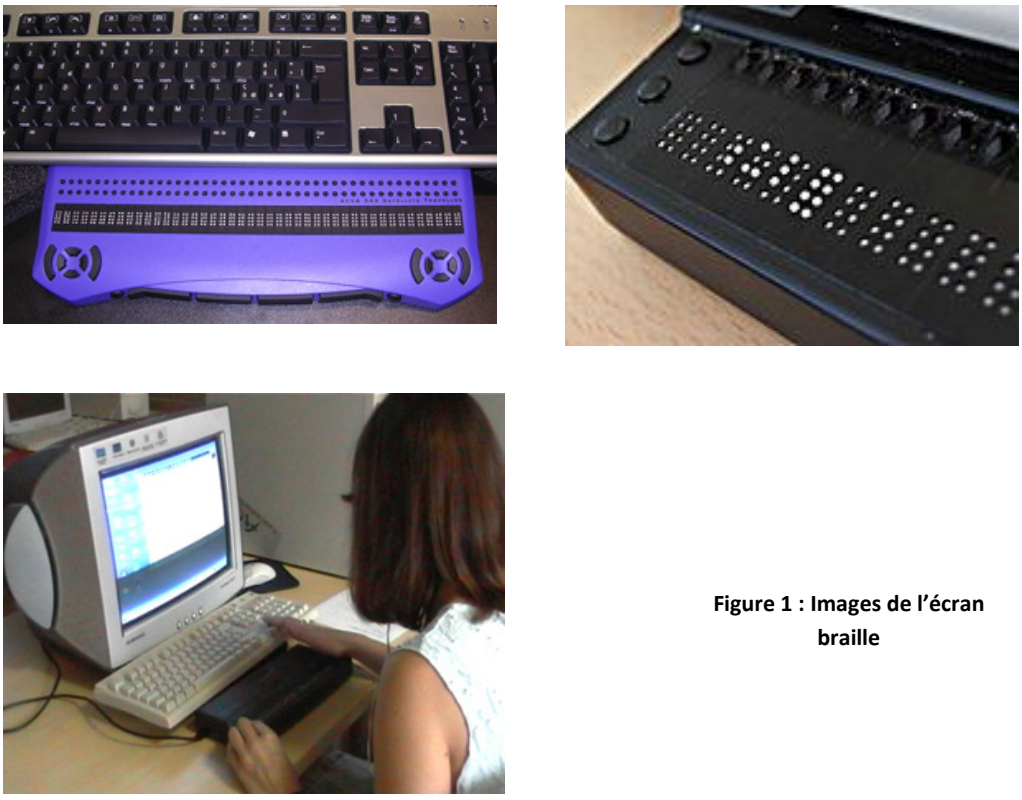


Figure 1 : Images de l'écran braille

Un autre produit à prendre en compte est le navigateur textuel Lynx (http://lynx.isc.org/lynx2.8.5/index.html). Grâce à son interface facilement compatible avec la synthèse vocale et la rapidité de chargement des pages, il est souvent utilisé par les aveugles à la place des navigateurs graphiques.

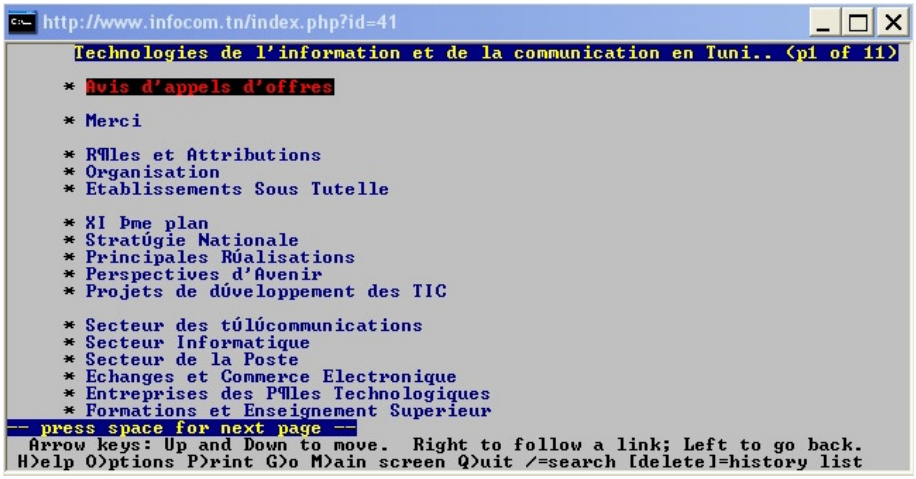

http://www.infocom.tn/index.php?id=41
Technologies de l'information et de la communication en Tuni.. (p1 of 11)
* Avis d'appels d'offres
* Merci
* R¶lles et Attributions
* Organisation
* Etablissements Sous Tutelle
* XI Þme plan
* Stratúgie Nationale
* Principales RÚalisations
* Perspectives d'Avenir
* Projets de dúveloppement des TIC
* Secteur des túlúcommunications
* Secteur Informatique
* Secteur de la Poste
* Echanges et Commerce Electronique
* Entreprises des P¶lles Technologiques
* Formations et Enseignement Superieur
-- press space for next page --
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list

Figure 2 : Page d'accueil de www.infocom.tn visualisée avec Lynx

Afin de contrôler l'ordinateur, les aveugles utilisent uniquement le clavier comme périphérique d'entrée.

II- Les personnes souffrant de troubles de l'audition

Les personnes souffrant de troubles, partiels ou complets, de l'ouïe n'ont pas besoin des outils d'assistance technologique et privilégient la souris.

Les difficultés que rencontrent ces personnes dans l'utilisation de l'ordinateur s'articulent en premier lieu autour de deux aspects :

- les contenus de type audio, qu'ils ne peuvent pas complètement percevoir ;
- les contenus textuels, qui dans certains cas peuvent être difficilement compréhensibles.

En ce qui concerne le premier aspect, l'emploi grandissant de présentations multimédia, les fichiers audio et vidéo dont la partie audio est essentielle, font que les personnes sourdes soient complètement exclues du contenu informatif.

Afin d'éviter de telles exclusions, il faut toujours fournir une alternative textuelle ou visuelle des contenus audio. Dans le cas d'une vidéo dans laquelle un présentateur parle, les alternatives possibles sont le texte ou les sous-titres de la vidéo ; dans le cas d'une alarme sonore, les alternatives sont un message textuel ou un changement de couleur.

Mais cela ne suffit pas ! Les textes doivent aussi être écrits de manière simple, car les sourds peuvent avoir d'importantes difficultés dans la compréhension de la langue.

En fait, cela vaut surtout pour les sourds de naissance si le défaut n'est pas vite diagnostiqué et que des prothèses ne sont pas implantées ; du fait du manque de retour auditif, ils peuvent se sentir

exclus du processus normal d'acquisition de la langue parlée et écrite, avec comme conséquences des limitations dans la connaissance de la langue maternelle. Le fait de connaître un nombre limité de mots, de structures verbales et de structures morpho-syntaxiques exclut ces personnes de la compréhension de ces textes qui s'expriment avec des phrases structurellement complexes.

La simplification du langage, même accompagnée d'une approche par icône et par images explicatives, est de toute manière, une qualité requise essentielle aussi pour d'autres types d'utilisateurs, comme les personnes souffrant de troubles cognitifs, les étrangers et les personnes avec un handicap culturel.

En ce qui concerne le deuxième aspect, il faut éviter pour les mêmes raisons les textes en mouvement. Les personnes sourdes, souffrant de troubles cognitifs, étrangères ou ayant un faible niveau de scolarisation peuvent avoir la nécessité de relire les contenus et, dans tous les cas, ils n'ont pas tous la même vitesse de lecture.

Enfin, il faut aussi noter qu'une partie des personnes sourdes communique à travers de la langue des signes. Il s'agit de vraies langues, différentes de pays en pays et parfois aussi au niveau régional, avec des règles propres et un vocabulaire riche.

Donc, en plus des textes simples, de l'utilisation d'icônes et d'images explicatives, au profit de tous les utilisateurs, la présence de vidéos de la traduction en langue des signes de quelques pages peut être fort appréciée par une bonne partie des personnes sourdes, surtout s'il y a de longs textes qui restent stables dans le temps.

Figure 3 : Interprète de la langue des signes

III- Les handicapés moteurs

Les handicapés moteurs dont il faut tenir compte au cours du développement des produits software ont des difficultés à contrôler leurs mains et donc à utiliser la souris et le clavier.

Les personnes paraplégiques, c'est-à-dire les personnes qui sont paralysées uniquement de la partie inférieure du corps, ne rencontrent pas de difficultés avec l'ordinateur, sinon de positionnement et d'ergonomie et par conséquent, ils ne seront pas pris en compte dans ce contexte.

Les problèmes dans l'usage des mains peuvent aller des tremblements jusqu'à la paralysie totale des membres supérieurs. Pour cette raison, les outils d'assistance technologique utilisés, qui concernent principalement la faculté d'insérer des données, peuvent être très différents selon la gravité de l'handicap. Ils consistent en des périphériques d'entrée comme des claviers spéciaux, des émulateurs de souris et de claviers, des commutateurs, des senseurs, des joysticks, des boules de commandes, des logiciels de reconnaissance vocale, jusqu'à des systèmes sophistiqués qui permettent d'entrer des données grâce aux battements des paupières ou au souffle dans une paille.

Normalement, si l'interface est utilisable avec le clavier, elle fonctionnera aussi avec les outils d'assistance technologique, mais il faut prendre en compte des dimensions et des espacements suffisants entre les objets sur l'interface afin que l'handicapé puisse « toucher » l'objet d'intérêt sans erreurs. Dans les images suivantes sont présentés quelques exemples de systèmes d'entrées pour handicapés moteurs.

Vu les difficultés que rencontrent les handicapés moteurs dans le contrôle de leurs mains, ils ont besoin de plus de temps pour exécuter les actions sur l'interface.

Pour cette raison, il est important qu'il n'y ait pas d'expirations de session inattendues ou de limites temporelles d'autres types.

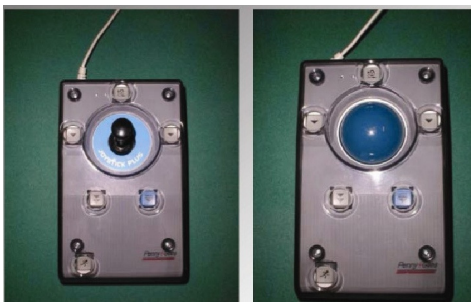
Pour ces mêmes raisons, il faut éviter les textes en mouvement, même si des mécanismes de blocage sont prévus.

Capteur de souffle

Émulateur de souris à souffle

Boutons à pression

Clavier spécial avec touches encastrées.



Joystick et boule de commande.

Figure 4 : Systèmes d'entrée pour handicapés moteurs

IV- Les personnes ayant des difficultés cognitives

Le domaine de troubles cognitifs est très vaste et diversifié. Dans ce contexte, on peut uniquement tenter de mettre en évidence, avec beaucoup de prudence, quelques aspects communs.

En général, ces personnes peuvent avoir des problèmes de concentration, d'attention, de lecture et de mémorisation qui, au niveau de l'interaction homme-machine, comportent des difficultés, par exemple, dans :

- la compréhension des pages complexes, tant au niveau du contenu que de la mise en page ;
- la lecture de texte en mouvement ;
- la lecture de textes sur une même page comportant des objets en mouvement (images, Flash, etc.)
- l'arrivée au bout de séquences d'actions complexes.

En général, les personnes souffrant de troubles cognitifs utilisent des aides et des outils d'assistance technologique qui permettent de procéder à des actions très immédiates et instinctives (comme l'écran tactile) ou des claviers simplifiés.

V- Les personnes ayant un handicap culturel

Font partie de cette catégorie les personnes âgées, les personnes ayant de faibles compétences informatiques, les personnes ayant une scolarisation inadéquate, les étrangers et les migrants qui ne connaissent pas bien la langue du pays d'accueil.

On peut comparer leurs exigences, par certains aspects, à ceux de certains types d'handicapés et elles se concentrent sur :

- la simplification du langage ;

- la simplification et la consistance/homogénéité de la mise en page ;
- la simplification de la séquence d'interaction (nombre d'étapes pour accomplir une action, nombre de clics pour arriver à la page désirée, etc.) ;
- l'utilisation d'icônes et d'images explicatives ;
- l'absence de textes et d'objets en mouvement ;
- l'indépendance du périphérique d'entrée.